

## **Comparação de modelos de previsão de séries temporais aplicados a bitcoin**

Gustavo Menezes de Sirqueira<sup>1\*</sup>; Daniel Alvarez Firmino<sup>2</sup>

<sup>1</sup> Aluno do curso de Data Science & Analytics do MBA USP/Esalq. Rua Moacyr Goulart Cunha Caldas, 60, 23a – Conceição; 09911-450 Diadema, São Paulo, Brasil

<sup>2</sup> Mestre em Ciências Econômicas/Economia, Orientador MBA Data Science & Analytics. Rua Alexandre Herculano, 120 – Vila Monteiro; 13418-445 Piracicaba, São Paulo, Brasil

\*autor correspondente: menezesgus@outlook.com

## Comparação de modelos de previsão de séries temporais aplicados a bitcoin

### Resumo

Este trabalho abordou a comparação de modelos de previsão de séries temporais aplicados ao preço do Bitcoin. O objetivo foi comparar o desempenho de modelos tradicionais de séries temporais, como ARIMA-GARCH, com modelos de redes neurais recorrentes, como LSTM e GRU. O Bitcoin, uma criptomoeda baseada em blockchain, apresenta alta volatilidade, o que justifica a busca por modelos capazes de prever seu comportamento. A pesquisa utilizou séries temporais do preço do Bitcoin e da taxa de juros americana como variável exógena, aplicando os modelos em dados de 2013 a 2024. As previsões foram avaliadas por meio das métricas RMSE e SMAPE, com resultados indicando que os modelos GRU apresentaram melhor desempenho nas janelas de previsão de curto prazo, enquanto o modelo ARIMA-GARCH demonstrou estabilidade, porém com menor precisão.

**Palavras-chave:** Bitcoin; ARIMA-GARCH; LSTM; GRU; Comparação de modelos de séries temporais.

### Introdução

Para se realizar transações entre 2 ou mais agentes no modelo financeiro tradicional, é necessária a existência de um intermediador, que por sua vez necessita de uma taxa para realizar a transação. Esse intermediário, usualmente bancos e instituições financeiras, mostra-se necessário para garantir a confiança, segurança e validade da transação. A moeda com a qual essa operação é realizada também precisa ser emitida por um órgão centralizador, como um banco central. Essa moeda muitas vezes tem seu lastro baseado em ouro ou confiança. Para abordar a questão mencionada acima, é preciso então a criação de um sistema que possa garantir transações seguras, confiáveis e válidas sem a necessidade de um intermediador e de um emissor centralizado (Patel et al., 2020; Vranken, 2017).

Em 2008 é proposto, no artigo “Bitcoin: A Peer-to-Peer Electronic Cash System” (Nakamoto, 2008), um conceito de moeda digital finita e transação online sem a necessidade de um intermediador, distribuído na rede e apoiado na tecnologia de “blockchain”, que por sua vez é baseada no artigo “How to time-stamp a digital document” (Haber e Stornetta, 1991). A tecnologia de transação atua como uma “caderneta pública” que é capaz de registrar movimentações em blocos, que por sua vez são encadeados a outros blocos gerados anteriormente, dando origem ao nome “blockchain” - cadeia de blocos. Devido ao caráter público da *blockchain*, existe transparência em suas transações. Um processo baseado em mecanismos de consenso e hash condicionados chamado de “mineração” garante que as transações válidas sejam adicionadas à corrente e, conseqüentemente, contabilizadas, além de garantir rastreabilidade e histórico. A mineração é realizada dentro dos nodos da rede e é um processo computacionalmente exigente e necessário para o funcionamento do sistema, gerando a necessidade de uma compensação aos agentes mineradores. Essa compensação é gerada pelo próprio sistema, e corresponde a uma quantia de bitcoin baseada na quantidade

de bitcoins existentes na rede. Dessa forma, a cada 210000 blocos o sistema diminui a recompensa na metade, o que é chamado de “halving”, tornando a operação de mineração gradativamente menos rentável (Vranken, 2017).

Desde o surgimento do bitcoin, o interesse pelas criptomoedas cresceu consideravelmente e atraiu a atenção de investidores e empresas. Sua tecnologia deu origem a outras criptomoedas, com enfoques diversos, variando na criação de moedas focadas em transações dentro de redes específicas até a criação de redes especializadas em transações, seguro e “supply chain” no campo da agricultura (Shen et al., 2020).

O mercado de criptomoedas corresponde, hoje, março de 2024, a 2,6 trilhões de dólares, sendo aproximadamente metade desse valor, 1,28 trilhões de dólares, correspondente ao bitcoin (Forbes Digital Assets, 2024). Alguns dos fatores determinantes para essa relevância foi a possibilidade da utilização do bitcoin na diversificação em portfólios e o potencial de geração de lucro, desvencilhado da centralização governamental. Entretanto, um maior potencial de lucro vem acompanhado de maior risco, e consequentemente maior potencial de perdas, em especial com posições de longo prazo, indicando uma alta volatilidade por parte do criptoativo (Nedved e Krisfoufek, 2023; Shen et al., 2020).

A previsão financeira desempenha um papel importante no mercado financeiro e no campo da economia, e é frequentemente utilizada na modelagem de ativos de alta volatilidade, na gestão de riscos e como ferramenta auxiliar na tomada de decisões. Alguns dos modelos mais comuns para esse fim são ARIMA para a modelagem e previsão da média e GARCH para lidar com a volatilidade, amplamente utilizados em artigos sobre a modelagem com retornos de bitcoin (Dyhrberg, 2015; Katsiampa, 2017; Phillip et al., 2017; Shen et al., 2020). Entretanto, a utilização de modelos de aprendizado de máquina não é tão amplamente difundida neste meio. Redes neurais recorrentes são capazes de aprender dependências de longo prazo em séries temporais, identificam características lineares e não lineares, apresentam maior sensibilidade a flutuações no tempo e podem preservar mais informação temporal durante o treino quando comparadas aos métodos econométricos tradicionais (Shen et al., 2020).

Levantados todos esses pontos, torna-se desejável a utilização de modelos capazes de lidar com a natureza imprevisível do bitcoin e que possuam características diferentes das já observadas em modelos comuns a literatura, tornando justificada a construção de um comparativo entre redes neurais recorrentes, em especial, GRU e LSTM, que possuem capacidade de lidar com o problema do gradiente desvanecente, e um modelo econométrico tradicional ARIMA-GARCH. Dessa forma, este trabalho foca na comparação de diferentes algoritmos de previsão aplicados ao valor do Bitcoin com o objetivo de avaliar a precisão e

desempenho de cada modelo, sem se aprofundar em uma análise exaustiva sobre a própria criptomoeda.

## **Material e Métodos**

### **Dados e Bibliotecas**

Para a realização da pesquisa foram utilizadas 2 séries temporais. A primeira tratou-se de uma série com o preço do bitcoin em dólar, obtida através do website CoinMarketCap.com (CoinMarketCap, 2024), referência em dados relacionados a criptomoedas. A segunda, que foi utilizada como variável exógena durante treino e previsões, tratou-se de uma série com o valor, em percentual, da taxa de juros americana, obtida através do website do Banco da Reserva Federal de ST. Louis (Federal Reserve Bank of St. Louis [FRED], 2024). A relação entre taxa de juros americana e criptomoedas, em especial o bitcoin, já foi atestada em outros trabalhos (Aboura, 2022; Dyhrberg, 2015), e existem indícios de que a mesma possa apresentar certa assimetria, na qual choques negativos na taxa de juros têm maior efeito sobre a moeda quando comparados a choques positivos (Köse e Ünal, 2023). Dessa forma, era esperado que a inclusão da variável pudesse potencializar a capacidade preditiva dos modelos. Ambas foram utilizadas em frequência mensal, e contemplaram o período de abril de 2013 até agosto de 2024, onde as últimas 10 entradas foram utilizadas para o teste da capacidade preditiva dos modelos. A validação se deu sobre o conjunto de treino. Todas as análises e construções de modelos foram realizados por meio da linguagem python e bibliotecas disponíveis, dentre elas, pandas e numpy para a manipulação dos dados, statsmodels e arch para a construção dos modelos ARIMA-GARCH, keras e tensorflow para a construção dos modelos LSTM e GRU e implementação da estratégia de random search, e scikit-learn para normalização dos dados, métricas de validação e no auxílio da validação no processo de treinamento.

### **Métodos**

Os métodos utilizados para previsão serão ARIMA em conjunto com GARCH, e as 2 variações mais conhecidas de RNN, sendo elas LSTM e GRU.

### **ARIMA**

Auto Regressive Integrated Moving Average [ARIMA], é uma técnica estatística comumente utilizada no campo das séries temporais para modelar dados que possuam dependência no tempo, como temperatura, valores de ações, PIB e afins. Conceitualmente, foca-se na captura de características como a sazonalidade, tendência e autocorrelação para a construção de uma análise e eventualmente a realização de previsões. ARIMA possui três principais componentes: Auto Regressive [AR] (Auto Regressivo), Integrated [I] (Integrado) e Moving Average [MA] (Média Móvel). AR refere-se a auto-regressão dos dados, ou seja, assume que o valor atual da série, é explicado por uma combinação linear dos valores passados, onde  $Y_{t-1}$  é capaz de explicar  $Y_t$ . A ordem desse componente, representado pela letra “p”, indica quantas defasagens utilizar na estimação do modelo. I refere-se ao número de diferenciações necessárias para tornar a série temporal estacionária. O modelo ARIMA exige estacionariedade para que sua estimação seja confiável e não viesada, dessa forma, é necessário que a série apresente média e variância constantes ao longo do tempo. A ordem desse componente, representada pela letra “d”, indica quantas diferenciações utilizar para se obter essas características. MA refere-se a média móvel dos erros residuais das previsões passadas. Dessa forma, as distâncias entre os valores previstos e valores reais dos períodos anteriores é incorporada ao modelo. A ordem desse componente, representado pela letra “q”, indica quantas defasagens de erros residuais utilizar na estimação do modelo (Box e Jenkins, 1968). Associado ao ARIMA são também empregados o teste de Dickey-Fuller Aumentado (Dickey e Fuller, 1979) para identificação de raiz unitária e determinação da ordem do componente I, e as funções de autocorrelação e autocorrelação parcial, para auxiliar na identificação das ordens dos componentes AR e MA.

## **GARCH**

Generalized Auto Regressive Conditional Heteroskedasticity [GARCH] (Bollorslev, 1986) ou GARCH foi proposto como uma extensão do modelo ARCH ou Auto Regressive Conditional Heteroskedasticity (Engle, 1982) para permitir uma estrutura mais flexível de defasagens. Enquanto ARCH especifica a variância condicional como uma função linear das variáveis amostrais, GARCH permite a inclusão de variâncias condicionais defasadas, tornando seu funcionamento análogo ao do modelo ARMA, porém aplicado à volatilidade. No campo das análises de séries temporais, temos a variância como uma representação da volatilidade. Preços de ações e outros ativos financeiros mudam ao longo do tempo, e apresentam variância inconstante, dessa forma, o modelo GARCH é capaz de captar características dessa dinâmica. O modelo apresenta dois componentes principais, sendo o primeiro o componente ARCH, auto regressivo, que utiliza a média ponderada das variâncias

passadas para prever a variância futura, e o componente GARCH, que considera o quadrado dos erros passados na regressão.

### Recurrent Neural Network [RNN]

As redes neurais recorrentes, ou RNNs, representam uma classe de redes neurais projetadas especificamente para processar dados sequenciais, permitindo a realização de previsões com base em informações sequenciais passadas. A principal característica das RNNs é a capacidade de reter informações de amostras anteriores por meio de um mecanismo de retroalimentação. Esse mecanismo re-insere a saída de uma amostra (estado oculto ou hidden state) como entrada para a próxima amostra, permitindo que as entradas anteriores influenciem diretamente as subsequentes.

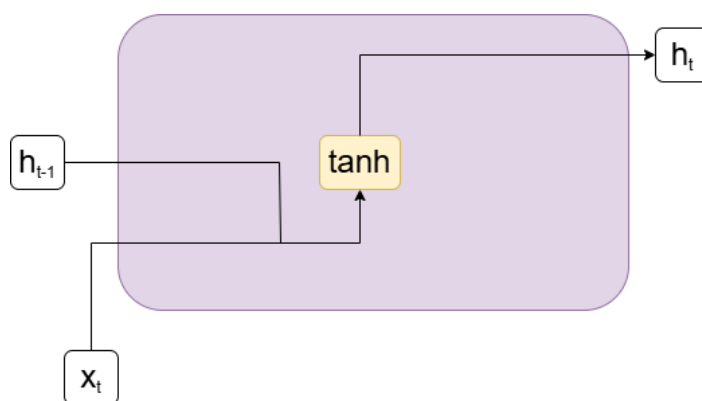


Figura 1. Arquitetura RNN  
Fonte: Elaborado pelo autor

No diagrama é possível observar como uma célula de RNN se comporta. Inicialmente, a célula recebe a entrada atual das variáveis exógenas, representada por  $X_t$ , juntamente com o estado oculto da célula anterior, representado por  $h_{t-1}$ . Os valores são então combinados e processados pela função de ativação tangente hiperbólica, representada aqui por  $\tanh$ , com o intuito de gerar um novo estado oculto, representado por  $h_t$ . Esse novo estado oculto é então a saída da célula, e poderá ser utilizado para a tarefa de predição e/ou como entrada para a próxima célula. Apesar de sua capacidade de processamento de dados sequenciais apresentar potencial, RNNs simples sofrem de duas limitações críticas: o problema do desaparecimento do gradiente e o problema da explosão do gradiente. Essas questões surgem devido ao comportamento dos pesos na rede neural que, quando diferentes de 1, multiplicam a entrada de maneira a aumentar ou diminuir seu valor exponencialmente ao longo do tempo. Conforme a sequência avança, e a saída transformada de uma amostra é utilizada como entrada para a amostra seguinte, esse valor é novamente multiplicado. Esse

processo pode levar a valores que tendem a zero (desaparecimento do gradiente) ou valores extremamente grandes (explosão do gradiente). Tais problemas prejudicam o cálculo eficiente do gradiente descendente, dificultando a obtenção dos parâmetros ótimos do modelo. Para mitigar esses problemas, várias técnicas e arquiteturas, como LSTM (Long Short-Term Memory) e GRU (Gated Recurrent Unit) foram desenvolvidas, proporcionando mecanismos que controlam o fluxo de informações e ajudam a manter a estabilidade do gradiente durante o treinamento.

### Long Short-Term Memory [LSTM]

O modelo Long Short-Term Memory, ou LSTM (Hochreiter e Schmidhuber, 1997) é uma arquitetura de rede neural recorrente projetada para lidar com dados sequenciais, mitigando problemas como o desaparecimento e a explosão do gradiente. Ele é construído com dois principais mecanismos de retenção de memória, sendo eles o estado da célula, que é responsável por armazenar informações de longo prazo, e o estado oculto, que mantém as informações de curto prazo. O estado da célula é isento de pesos e vieses, permitindo a preservação da memória ao longo do tempo. Em seu funcionamento, são utilizadas três portas, responsáveis pelo fluxo de informações através da rede e do tempo. As portas, por sua vez, são controladas por funções de ativação sigmóide.

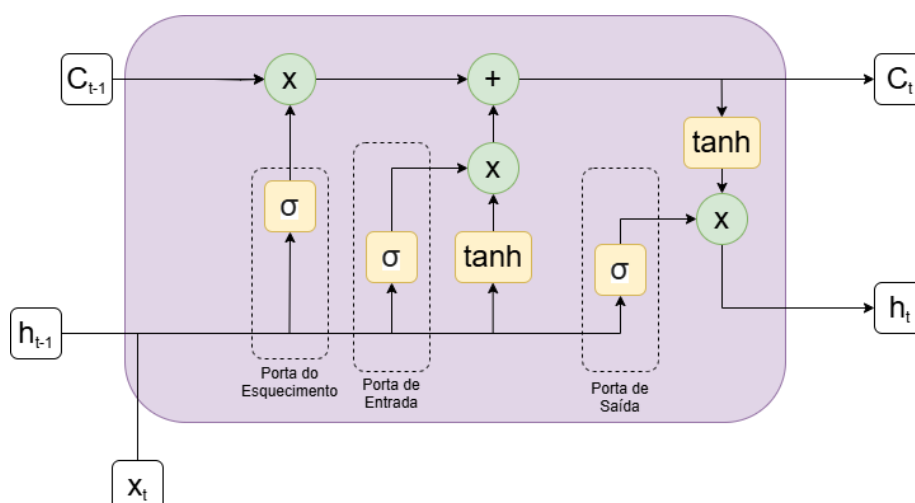


Figura 2. Arquitetura LSTM  
Fonte: Elaborado pelo autor

No diagrama é possível observar como a célula LSTM se comporta. Inicialmente, assim como na arquitetura RNN simples, a célula recebe a entrada atual das variáveis exógenas, representada por  $x_t$ , juntamente com o estado oculto da célula anterior, representado por  $h_{t-1}$ . Além desses valores, há também a entrada de  $C_{t-1}$ , que representa a

memória de longo prazo (ou estado da célula). Os valores de  $X_t$  e  $h_{t-1}$  são então combinados e inseridos em 3 portas. A primeira, porta do esquecimento, tem como função determinar o que deve ser esquecido da memória de longo prazo  $C_{t-1}$ . A segunda, a porta de entrada, é responsável por decidir quais novas informações serão adicionadas a essa memória, e em conjunto com a função tangente hiperbólica, qual proporção, incidindo na determinação de  $C_t$ , a nova memória de longo prazo (ou estado da célula) que será passada para a próxima célula. A terceira, a porta de saída, é responsável pela regulação da memória de curto prazo e determina quais informações serão utilizadas para se calcular o novo estado oculto. Essa informação é então combinada com a transformação via tangente hiperbólica da memória de longo prazo, dando origem ao estado oculto da unidade LSTM, representada por  $h_t$ , que poderá ser utilizado para a tarefa de predição e/ou como entrada para a próxima célula. Dessa forma, esses mecanismos combinados permitem que o modelo LSTM aprenda e mantenha relações temporais complexas em sequências de dados.

### Gated Recurrent Unit [GRU]

Similar à LSTM, a GRU ou Gated Recurrent Unit (Chung et al., 2014) é projetada para lidar com dados sequenciais, sendo essencialmente uma simplificação da LSTM.

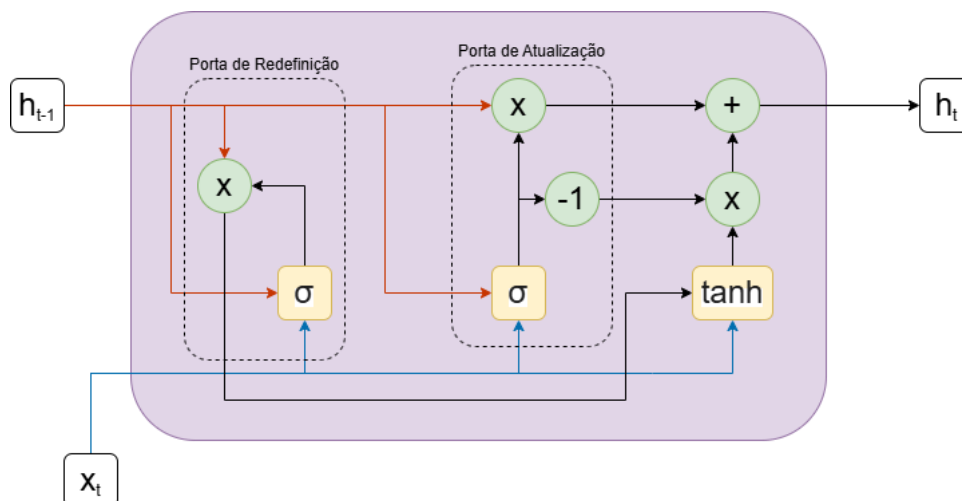


Figura 3. Arquitetura GRU  
Fonte: Elaborado pelo autor

No diagrama apresentado, é possível observar o funcionamento desta arquitetura. Primeiro, a célula recebe o estado oculto da célula anterior,  $h_{t-1}$ , assim como os valores das variáveis exógenas atuais,  $x_t$ . Ambas então passam pela porta de redefinição, que é responsável por controlar qual a proporção de informações de longo prazo a serem esquecidas, transformando e combinando as variáveis de entrada. Posteriormente,  $h_{t-1}$  e  $x_t$



são novamente combinadas e processadas na porta de atualização, responsável pela determinação do quanto de informação passada será utilizada para atualizar o novo estado oculto. Por fim, é calculado um candidato ao novo estado oculto utilizando a função tangente hiperbólica juntamente com uma combinação linear de  $X_t$  e o produto resultante da porta de redefinição. Esse candidato é então combinado com o produto resultante da porta de atualização, dando origem ao novo estado oculto,  $h_t$ . Devido à sua arquitetura mais simples, em comparação com a LSTM, a GRU tende a resultar em um treinamento mais rápido.

### **Estratégia Empírica**

Primeiramente, as séries temporais de preços do bitcoin e taxa de juros americana foram analisadas e decompostas, com o intuito de se observar características das séries, como suas movimentações ao longo do tempo, tendências, sazonalidades, e a presença de heterocedasticidade. Posteriormente, as séries serão separadas em 2 conjuntos, um de treino, contendo 127 observações, e um de teste, contendo as últimas 10. Os dados foram normalizados antes de serem imputados nos modelos LSTM e GRU, a fim de facilitar o treinamento.

Para a construção do modelo conjunto ARIMA-GARCH, foram levados em consideração os achados da decomposição, assim como a aplicação de testes para identificação de raiz unitária e estacionariedade. Inicialmente foi estimado o modelo ARIMA, que teve seus resíduos extraídos e utilizados na estimação do modelo GARCH. Esta metodologia para a união de modelos ARIMA e GARCH é comumente utilizada por unir a capacidade de previsão da média do modelo ARIMA com a flexibilidade ante a variância proporcionado pelo modelo GARCH (Mohammadi e Su, 2010; Narendra e Eswara, 2015). Por fim, foram realizadas as previsões de média e variância esperadas por meio dos modelos, e os valores foram unidos em uma soma para determinar a previsão de fato a ser comparada.

A construção dos modelos LSTM e GRU foi realizada de maneira similar entre eles. A bibliografia relevante indica que é recomendada a utilização de redes com mais de uma camada, assim como camadas de dropout, capazes de desativar aleatoriamente certas proporções de neurônios na rede com o intuito de minimizar o “overfitting”. Com isso, no modelo LSTM será utilizada uma camada LSTM, uma camada de dropout, outra camada LSTM, outra camada de dropout, e uma camada densa para saída. Similarmente, o modelo GRU possuirá uma camada GRU, uma camada de dropout, outra camada GRU, outra camada de dropout e uma camada densa para saída. Com isso, os modelos serão capazes de apresentar profundidade e maior potencial de performance (Ferdiansyah et al., 2019; Hamayel et al., 2021; Lin, 2022; Seabe et al., 2023; Sudhakaran et al., 2024).

Os hiperparâmetros serão selecionados por meio do método de random search, indicado para esse tipo de problema por apresentar performance potencialmente superior a seleção manual ou grid search (Bergstra e Bengio, 2012). O método funciona selecionando aleatoriamente diferentes combinações de parâmetros ou soluções, múltiplas vezes, e avaliando essas escolhas, dado um espaço de busca, que é apresentado na tabela a seguir:

Tabela 1. Hiperparâmetros e valores possíveis

Hiperparâmetro	Valores possíveis
Unidades	20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250
Taxa de “dropout”	0.2, 0.3, 0.4, 0.5
Taxa de aprendizagem	0.01, 0.001, 0.0001
Tamanho do lote	32, 64, 96, 128
Épocas	25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300

Fonte: Dados originais da pesquisa

A escolha dessa seleção de valores se deu de forma a oferecer variações significativas de hiperparâmetros a serem permutados, ao passo de tornar o processamento e eventuais reprocessamentos viáveis em máquinas caseiras e virtuais, como as disponíveis no Google Colab. Os possíveis valores de unidades e de taxas de dropout estão alinhados com a necessidade de experimentação de diferentes capacidades e complexidades do modelo, equilibrando “underfitting” e “overfitting” (subajuste e sobreajuste) (Bergstra e Bengio, 2012; Srivastana et al., 2014). Múltiplas taxas de aprendizagem são essenciais para equilibrar velocidade de convergência e estabilidade no treinamento, enquanto lotes menores são indicados para arquiteturas sofisticadas como LSTM e GRU, dessa forma, os valores escolhidos permitem um grau flexibilização sem detrimento da estabilidade (Bergstra et al., 2011). Por último, a seleção das épocas permite a determinação de um ponto de parada apropriado, evitando também o “overfitting”, a medida que oferece alto grau de variação (Bengio et al., 1994).

O otimizador utilizado será o “adaptive moment estimation” [adam] e a função de custo o erro quadrático médio, ambos indicados para séries temporais e para as arquiteturas LSTM e GRU (Sudhakaran et al., 2024). Além disso, foram consideradas quatro janelas de tempo, sendo elas, cinco, dez, vinte e quarenta períodos, resultando em um total de oito modelos, quatro de arquitetura LSTM e quatro de arquitetura GRU. Também é utilizado o método “TimeSeriesSplit” presente no pacote scikit-learn a fim de permitir validação durante o processo de busca. É utilizada a semente 42 para determinação dos estados aleatórios em código, a fim de possibilitar a replicação do procedimento aqui descrito.

Construídos os modelos de deep learning, será realizada a previsão para os dez períodos contidos no conjunto de testes.

As previsões realizadas por meio dos três métodos serão então comparadas por meio da utilização das métricas RMSE - Root mean square error ou Raiz do erro quadrático médio - e sMAPE - Symmetric mean absolute percentage error ou Erro percentual absoluto médio simétrico. A métrica RMSE penaliza grandes erros, pois em seu processo de cálculo, eleva o erro ao quadrado antes de extrair a raiz, tornando-o bastante sensível a outliers. Essa métrica apresenta valores na mesma unidade de medida dos dados originais, o que facilita a interpretação. Além disso, normaliza erros positivos e negativos, oferecendo uma visão equilibrada dos desvios. No entanto, o RMSE tem limitações significativas, especialmente ao lidar com dados de diferentes magnitudes, o que pode distorcer a análise comparativa em séries temporais com escalas variadas.

Por outro lado, sMAPE é uma métrica normalizada que apresenta valores em percentual, permitindo a comparação entre séries temporais de diferentes magnitudes, independentemente de suas unidades de medida. Sendo uma métrica simétrica, o sMAPE penaliza igualmente desvios positivos e negativos, abordando as fraquezas do RMSE. A combinação do uso de RMSE e sMAPE proporciona uma análise mais completa, robusta e confiável, pois aborda diferentes aspectos do erro, garantindo uma avaliação mais equilibrada e abrangente do desempenho dos modelos de previsão (Goodwin e Lawton, 1999; Hyndman e Koehler, 2006; Shcherbakov et al., 2013). Dessa forma, os modelos foram comparados e assim foi determinado qual apresenta melhor performance aplicada aos dados tratados no trabalho.

## **Resultados e Discussão**

As séries de preço de bitcoin e taxa de juros americana apresentam a seguinte representação gráfica:

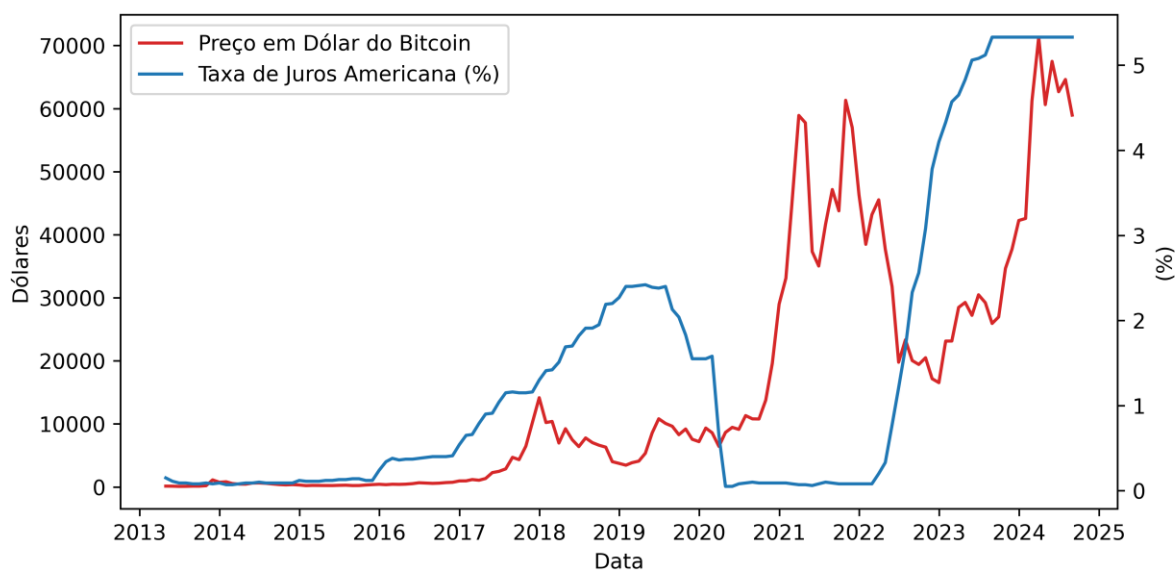


Figura 4. Gráfico comparativo entre preço em dólar do bitcoin e taxa de juros americana  
Fonte: Dados originais da pesquisa

Analisando a figura é possível se observar que em diversos momentos as movimentações das séries foram similares, ainda que em magnitudes diferentes. Em especial, ambas apresentaram subidas entre o período de 2016 a 2020, assim como entre o período de 2022 adiante. A ascensão do bitcoin a partir de 2017 é bastante atribuída ao aumento da cobertura midiática e dos interessados em investimento de modo geral (Kayal e Rohilla, 2021). Entretanto, é importante observar que entre 2020 e 2022 essa dinâmica se inverte, e é possível notar um aumento do preço do bitcoin à medida que a taxa de juros se mantém baixa. Isso se deve a pandemia de COVID19 e as políticas adotadas pela Reserva Federal americana a fim de manter níveis de emprego e atividade econômica estável. O mesmo não aconteceu com o bitcoin, que, em um momento de incerteza como o de uma pandemia, se mostrou como um ativo “porto seguro” na visão de muitos investidores, resultando em um aumento considerável do seu valor frente ao dólar (Nguyen, 2022; Kumar e Padakandla, 2022).

Abaixo se encontram as decomposições de ambas as séries, tanto pelo método aditivo quanto pelo método multiplicativo:

## Preço em Dólar do Bitcoin - Tendência

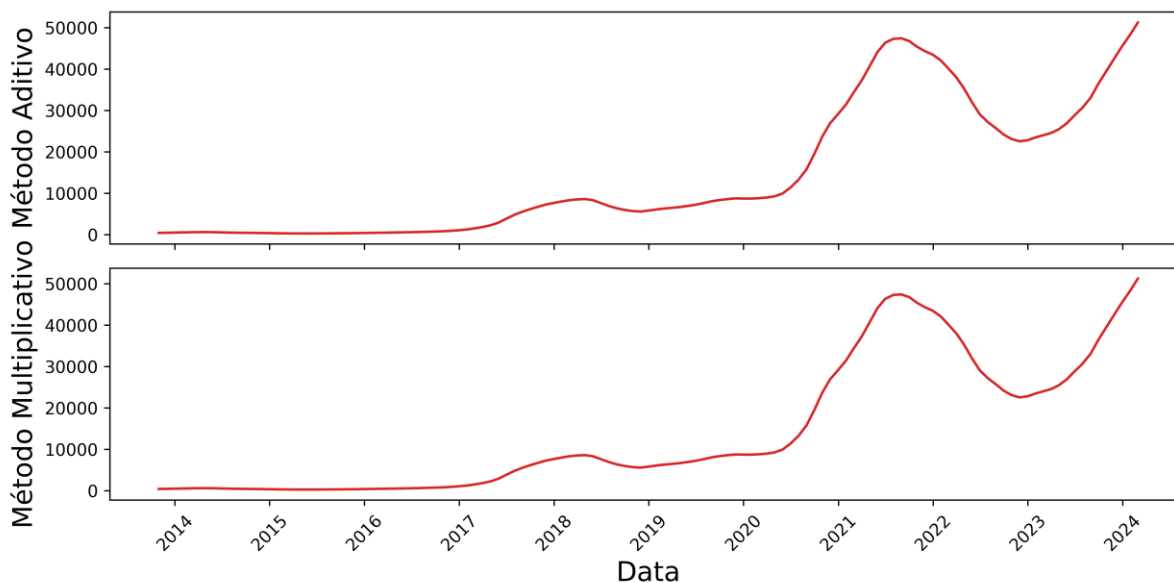


Figura 5. Gráficos de tendência do preço em dólar do bitcoin  
Fonte: Dados originais da pesquisa

A decomposição da série de preço do bitcoin apresenta tendência significativa a partir de 2017 sob ambos os métodos, o que aparenta coincidir com o aumento de sua popularidade. A tendência também mostra um possível aumento no impacto da variação do preço a medida que a moeda se torna mais popular e valiosa, mostrando uma possível alta sensibilidade ao efeito manada, como já atestado em outros trabalhos (Qezelbash, 2023), dessa forma, é possível justificar a necessidade de diferenciação da série ao utilizar o modelo ARIMA.

## Preço em Dólar do Bitcoin - Sazonalidade

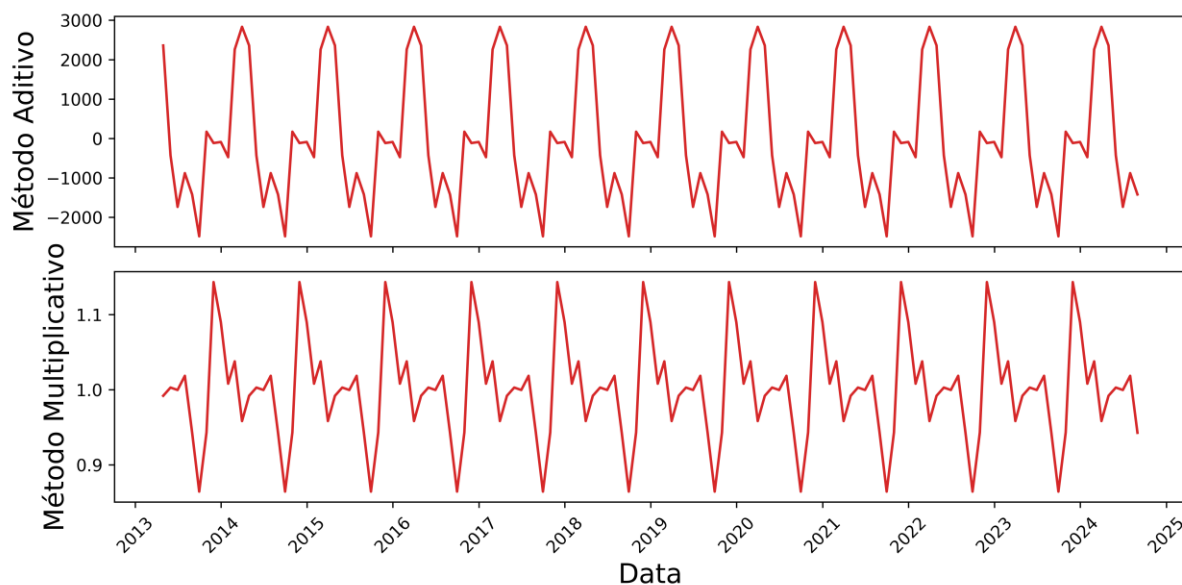


Figura 6. Gráficos de sazonalidade do preço em dólar do bitcoin  
Fonte: Dados originais da pesquisa

Com relação a sazonalidade, é observável que, apesar de presente, sua magnitude não parece ser proporcionalmente alta em ambos os métodos. Múltiplos artigos encontraram fraca presença de sazonalidade, ou, quando significativa, usualmente presente em uma frequência diária e não mensal (Fraz et al., 2019; Kaiser, 2019; Long et al., 2020; Shanaev e Ghimire, 2022). Desse modo, os efeitos encontrados alinham-se com o que é apresentado na maior parte da bibliografia.

## Preço em Dólar do Bitcoin - Resíduos

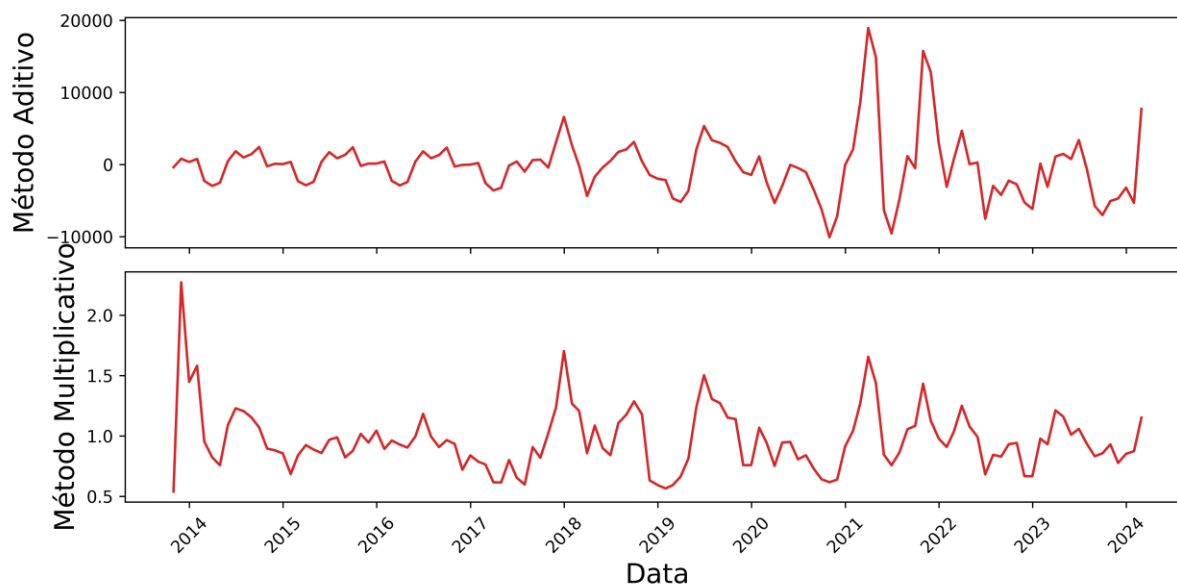


Figura 7. Gráficos de resíduos do preço em dólar do bitcoin  
Fonte: Dados originais da pesquisa

Com relação aos resíduos, é notável um forte sinal de existência de heterocedasticidade sob ambos os métodos, implicando em uma provável inconstância de variância ao longo da série. Essa inconstância justifica o emprego do modelo GARCH.

## Taxa de Juros Americana - Tendência

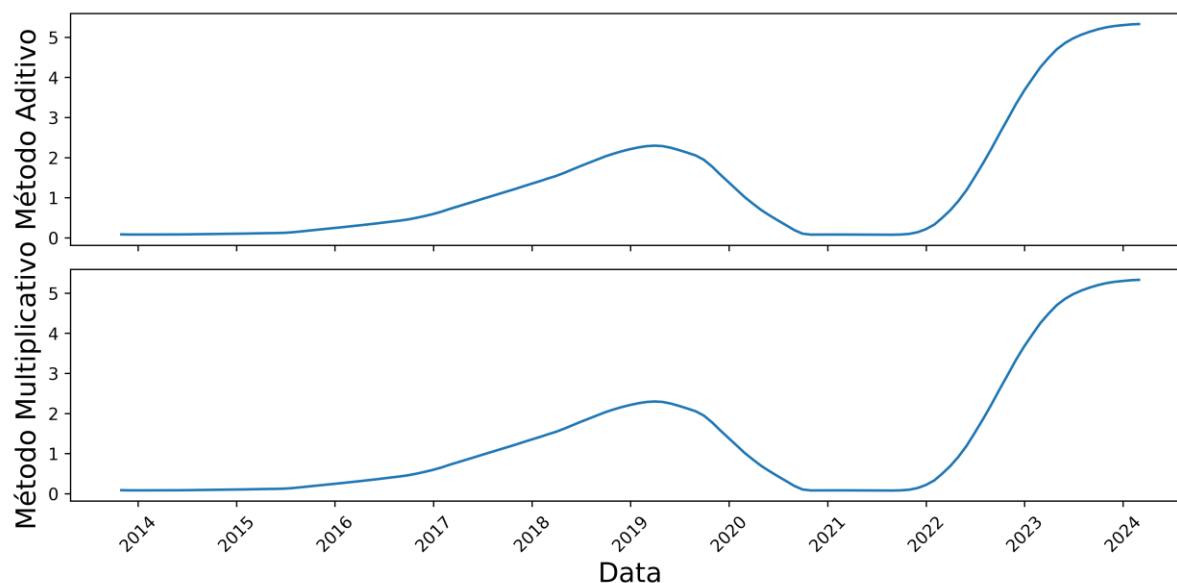


Figura 8. Gráficos de tendência da taxa de juros americana  
Fonte: Dados originais da pesquisa

A decomposição da taxa de juros americana também apresenta tendência, em especial, nos períodos anterior e posterior a pandemia de COVID19, onde é possível notar uma movimentação bastante similar à encontrada na série de preço em dólar do bitcoin.

### Taxa de Juros Americana - Sazonalidade

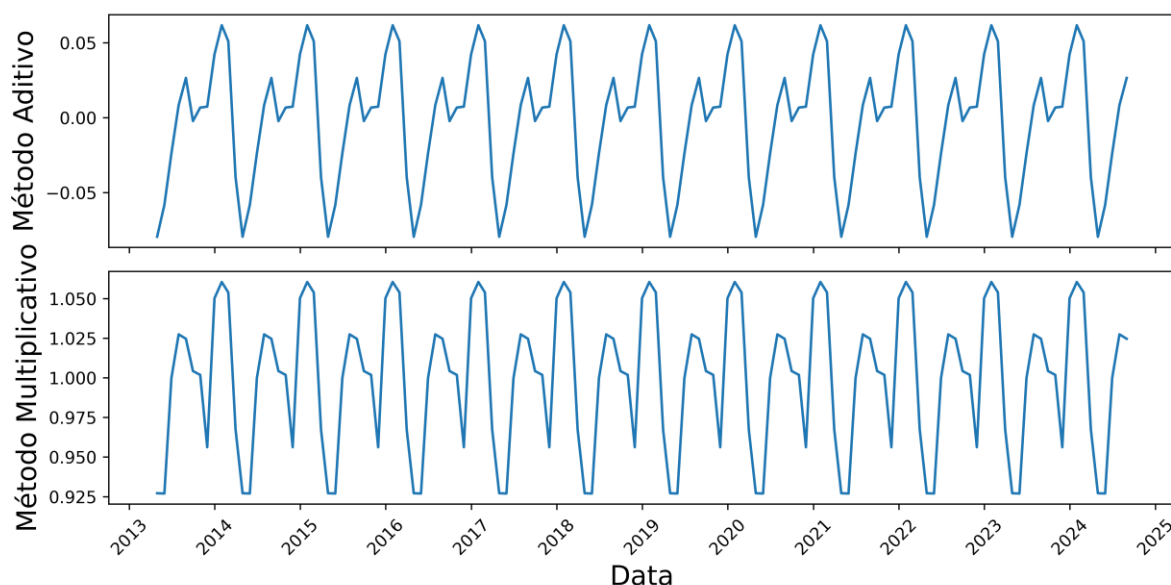


Figura 9. Gráficos de sazonalidade da taxa de juros americana

Fonte: Dados originais da pesquisa

Com relação a sazonalidade, é possível notar pouca influência sobre a série, indicando que essa característica é fraca ou inexistente.



## Taxa de Juros Americana - Resíduos

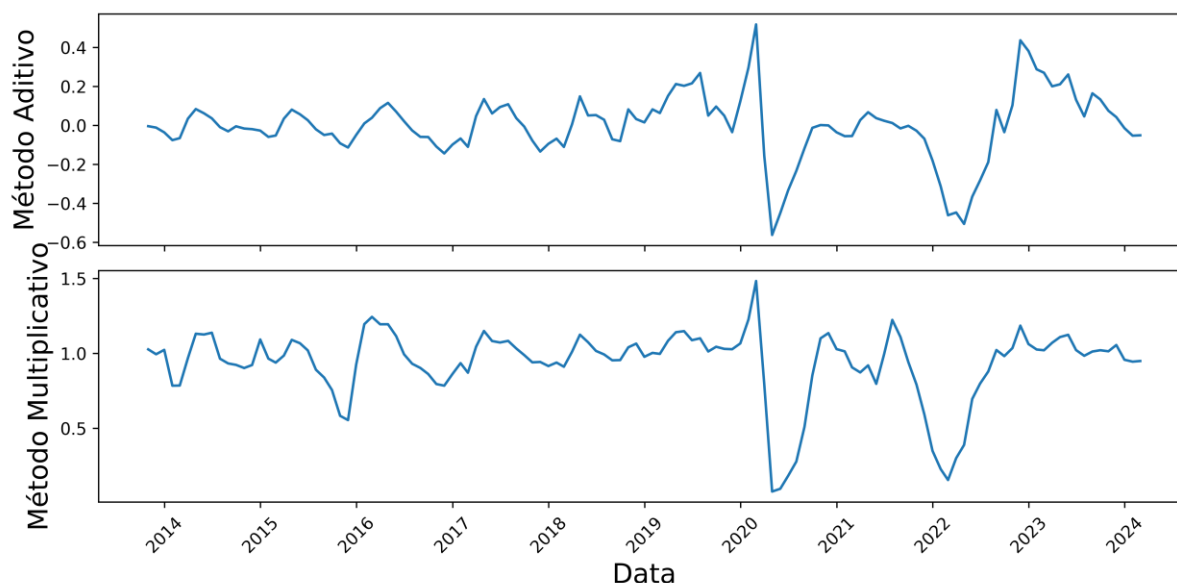


Figura 10. Gráficos de resíduos da taxa de juros americana  
Fonte: Dados originais da pesquisa

Os resíduos também apresentam um forte sinal de existência de heterocedasticidade sob ambos os métodos, implicando em uma provável inconstância de variância ao longo da série. É importante ressaltar que, para ambas as decomposições, existem limitações que podem influenciar os dados e análises obtidas. Uma quantidade alta de ruído, alterações de sazonalidade ao longo do tempo e restrições do algoritmo utilizado para a decomposição podem ter forte influência nos resultados obtidos.

Realizada a análise e decomposição das séries, os dados foram divididos em um conjunto de treinamento, com as primeiras 127 observações, e um conjunto de testes, com as últimas 10. Com o intuito de facilitar o treinamento dos modelos LSTM e GRU, os dados são normalizados antes de serem imputados nos modelos. Já para o modelo ARIMA-GARCH, a escala é mantida.

A construção do modelo ARIMA-GARCH se dá seguindo o que foi estabelecido na seção referente a estratégia empírica, com o emprego do teste de Dickey-Fuller Aumentado (Dickey e Fuller, 1979), identificando a presença de raiz unitária e não estacionariedade na série temporal. Uma vez diferenciada, a série tornou-se estacionária de acordo com o teste. Entretanto, visualmente, era ainda possível notar a existência de variância inconstante, forte indicativo de existência de heterocedasticidade, justificando o emprego do modelo GARCH.

Obtida a ordem de integração, foi construído diversos modelos com ordens AR e MA variando entre 1 e 5. Para cada variação era então realizada uma análise de resíduos por meio das funções de autocorrelação e autocorrelação parcial. O objetivo era encontrar o

modelo com a menor ordem que não apresentasse resíduos autocorrelacionados. Por meio desta estratégia, foi encontrado o modelo de ordem ARIMA(2,1,2). A ordem encontrada foi reforçada ao se utilizar também do método de auto-arima, método esse que itera por diversas ordens, buscando otimizar o indicador AIC.

Novamente foi executado o teste de Dickey-Fuller Aumentado, sobre os resíduos, obtendo-se o indicativo de estacionaridade. Foram construídos múltiplos modelos com ordens GARCH e ARCH variando entre 1 e 5, utilizando como variável endógena os resíduos obtidos do modelo ARIMA. O mesmo processo utilizado para definição das ordens foi realizado, porém, não foi obtida diferença significativa na análise dos resíduos, indicando a escolha do modelo GARCH(1,1), que apresentava mesma performance que os de ordens maiores. Dessa forma, obteve-se o modelo conjunto ARIMA-GARCH com as ordens (2,1,2)-(1,1).

Tabela 2. Modelo ARIMAX (2,1,2)

Variável	Coeficiente	Erro Padrão	P Valor
Taxa de Juros Americana	-2278,3069	2305,498	0,323
AR Lag 1	-0,3668	0,12	0,002
AR Lag 2	-0,7519	0,099	0
MA Lag 1	0,5882	0,137	0
MA Lag 2	0,7096	0,135	0
$\sigma^2$	1,60E+07	0,099	0

Fonte: Resultados originais da pesquisa

Apesar da utilização da variável “taxa de juros americana” no modelo, a mesma apresenta insignificância dentro dos níveis habituais de 1%, 5% e 10%.

Tabela 3. Modelo GARCH(1,1)

Variável	Coeficiente	Erro Padrão	P Valor
$\omega$	2,85E+05	5,13E+04	2,60E-08
$\alpha_1$	0,4219	6,40E-02	4,29E-11
$\beta_1$	0,5781	6,82E-02	2,43E-17

Fonte: Resultados originais da pesquisa

Por meio do processo de random search, foram encontradas as configurações dos modelos LSTM e GRU:

Tabela 4. Arquitetura dos modelos via random search

Célula	Janela	Unidades	Taxa de Dropout	Taxa de Aprendizagem	Tamanho do Lote	Épocas
LSTM	5	100	0,4	0,01	128	125
LSTM	10	50	0,4	0,01	96	150
LSTM	20	100	0,4	0,001	128	125
LSTM	40	50	0,3	0,01	32	50
GRU	5	100	0,2	0,001	128	75
GRU	10	125	0,4	0,001	64	75
GRU	20	100	0,2	0,001	128	75
GRU	40	100	0,2	0,001	128	75

Fonte: Resultados originais da pesquisa

## Previsões

Com os modelos dimensionados se deram as previsões, que podem ser observadas de maneira visual abaixo:

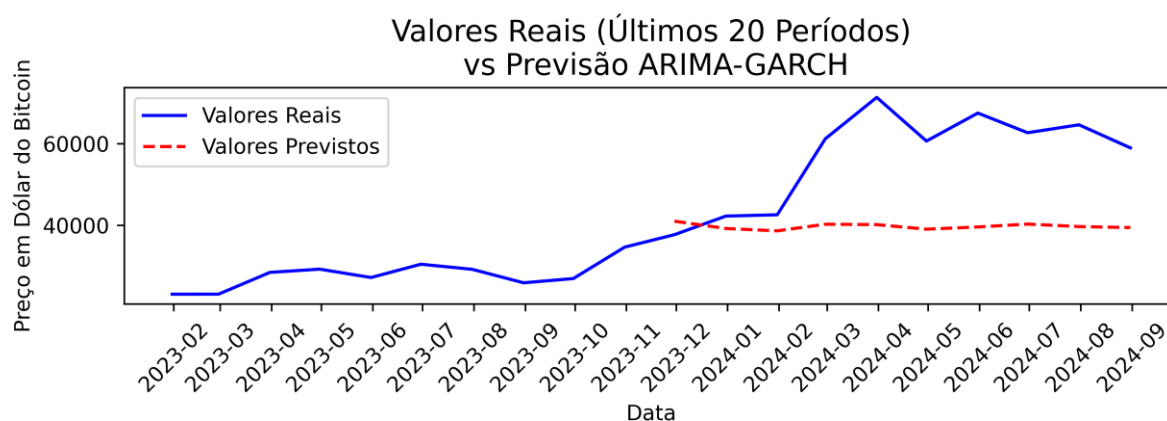


Figura 11. Gráfico comparativo previsto vs real ARIMA-GARCH

Fonte: Dados originais da pesquisa

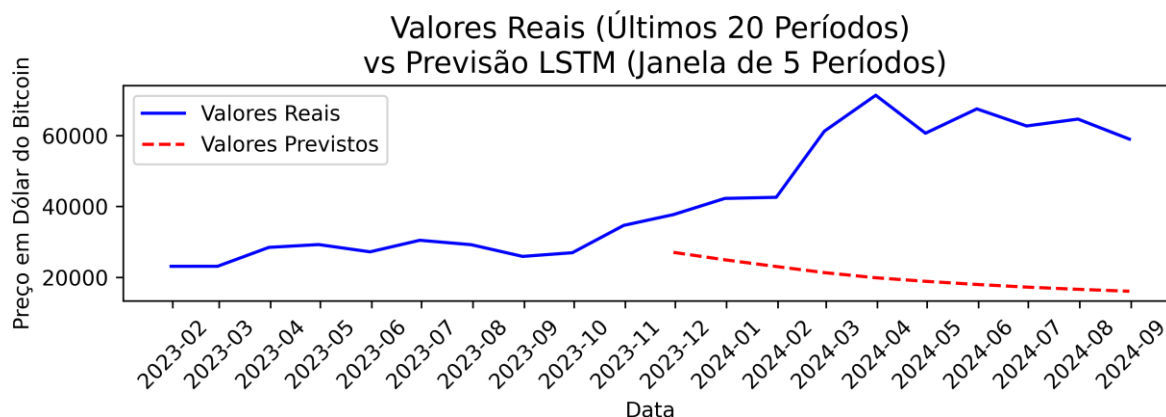


Figura 12. Gráfico comparativo previsto vs real LSTM janela de 5 períodos  
Fonte: Dados originais da pesquisa

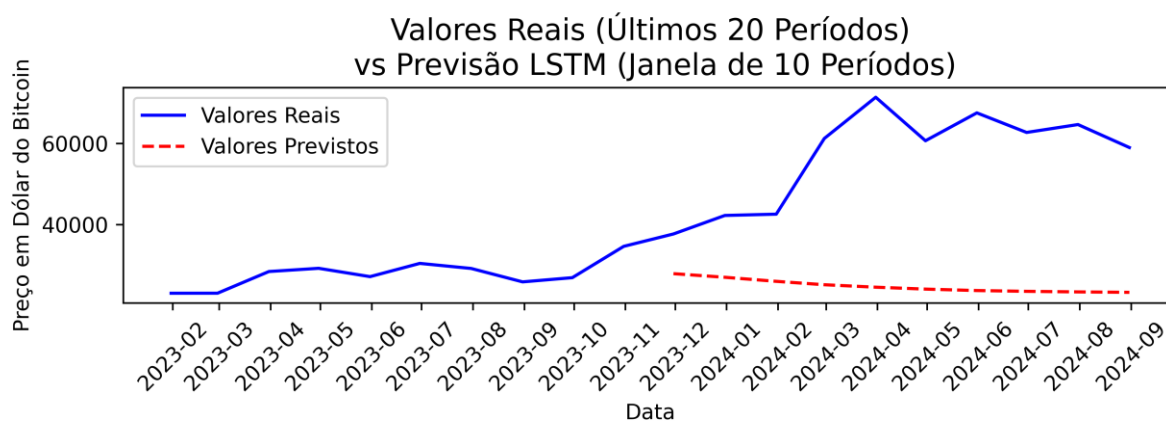


Figura 13. Gráfico comparativo previsto vs real LSTM janela de 10 períodos  
Fonte: Dados originais da pesquisa

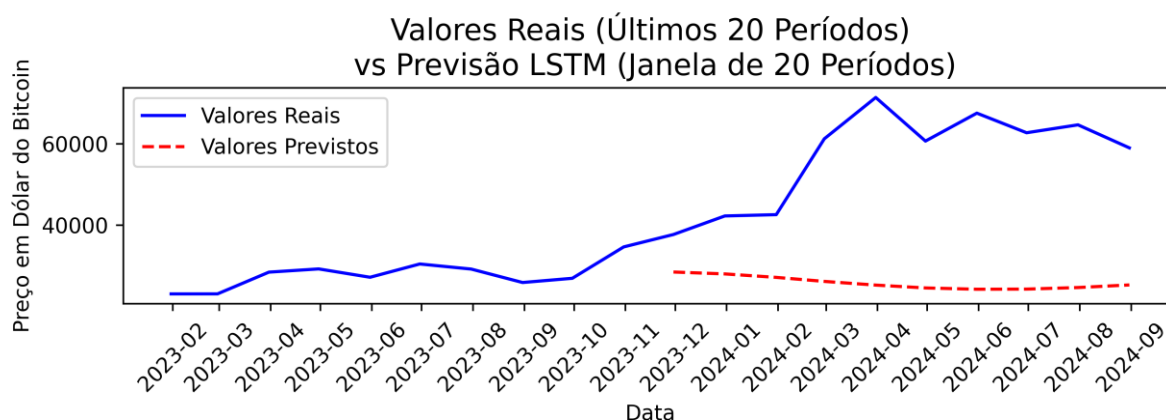


Figura 14. Gráfico comparativo previsto vs real LSTM janela de 20 períodos  
Fonte: Dados originais da pesquisa

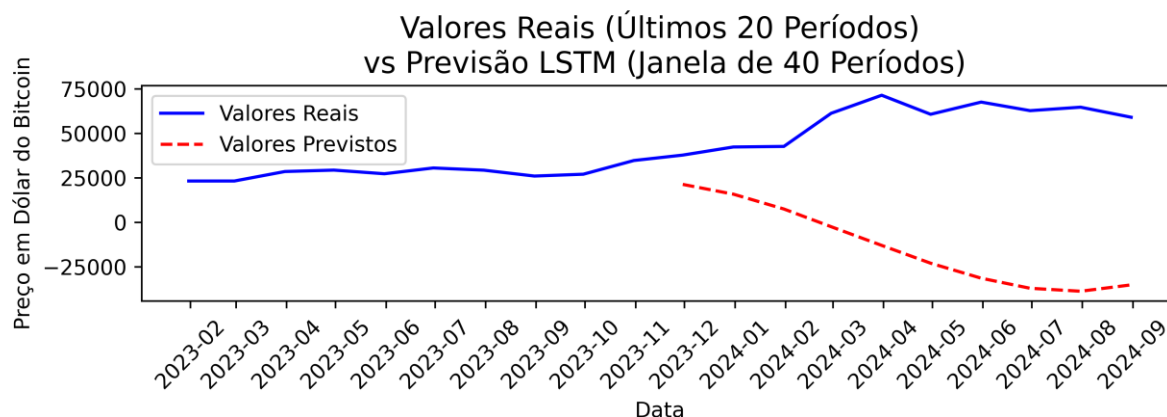


Figura 15. Gráfico comparativo previsto vs real LSTM janela de 40 períodos  
Fonte: Dados originais da pesquisa

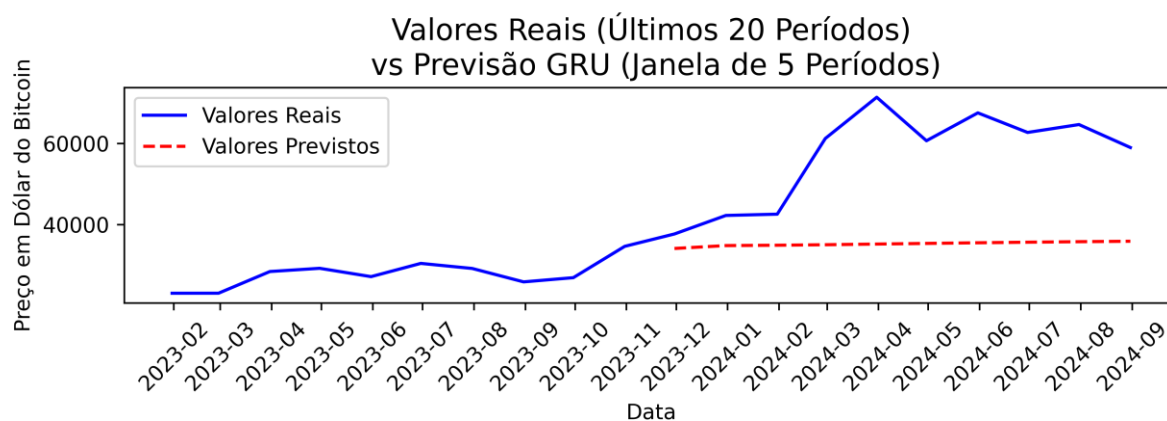


Figura 16. Gráfico comparativo previsto vs real GRU janela de 5 períodos  
Fonte: Dados originais da pesquisa

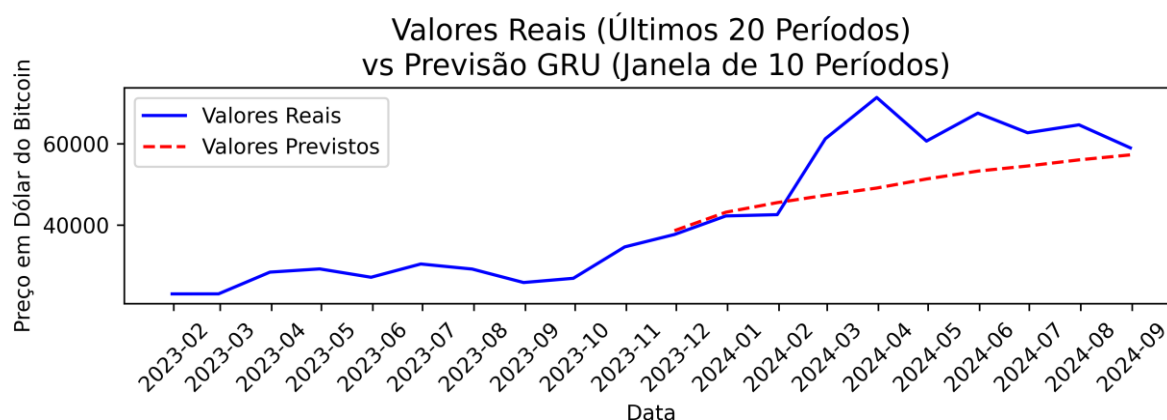


Figura 17. Gráfico comparativo previsto vs real GRU janela de 10 períodos  
Fonte: Dados originais da pesquisa

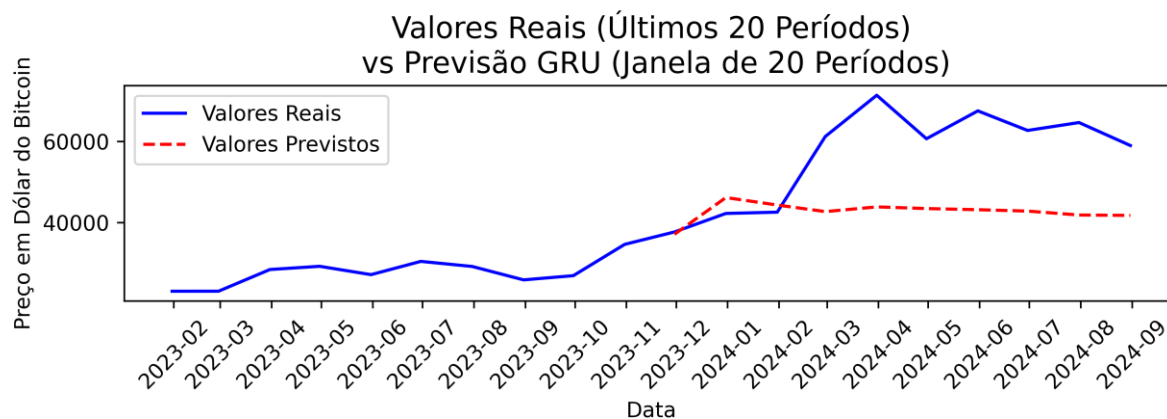


Figura 18. Gráfico comparativo previsto vs real GRU janela de 20 períodos  
Fonte: Dados originais da pesquisa

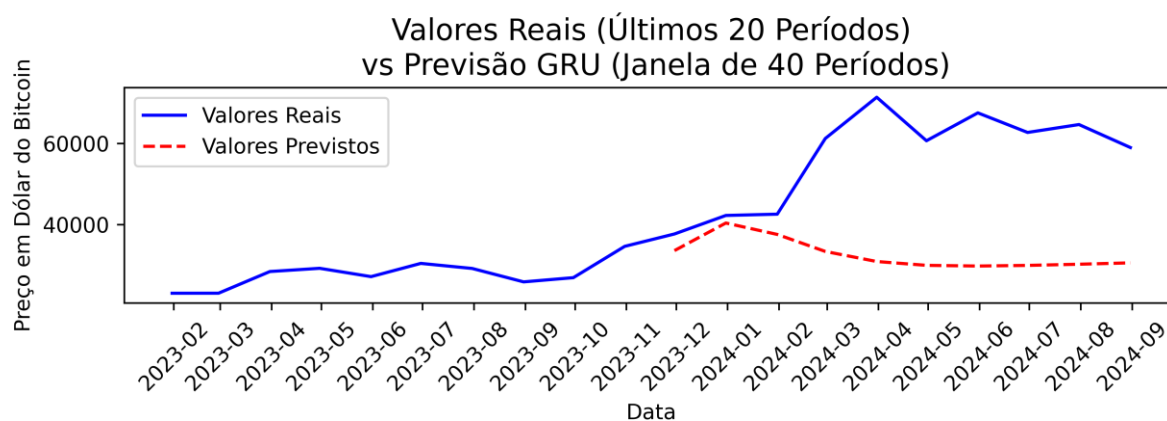


Figura 19. Gráfico comparativo previsto vs real GRU janela de 40 períodos  
Fonte: Dados originais da pesquisa

Na tabela abaixo é possível observar o resultado da comparação entre os modelos utilizando as métricas RMSE e sMAPE:

Tabela 5. Performance dos modelos

Modelo	RMSE	sMAPE
Previsão ARIMA-GARCH	20470,68272	34,89335347
Previsão LSTM (Janela de 5 Períodos)	39252,72054	91,91452952
Previsão LSTM (Janela de 10 Períodos)	34355,70298	75,45948516
Previsão LSTM (Janela de 20 Períodos)	33510,48607	72,38148643
Previsão LSTM (Janela de 40 Períodos)	77267,10151	168,7873815
Previsão GRU (Janela de 5 Períodos)	24206,055	44,62905862
Previsão GRU (Janela de 10 Períodos)	10622,66617	14,50398544
Previsão GRU (Janela de 20 Períodos)	17901,89902	28,89072889
Previsão GRU (Janela de 40 Períodos)	28034,02861	51,74115612

Fonte: Resultados originais da pesquisa

Os resultados mostraram que o modelo ARIMA-GARCH teve um desempenho intermediário, com um RMSE elevado, o que indica que suas previsões estavam relativamente distantes dos valores reais. No entanto, o sMAPE sugere que a variação percentual média dos erros foi mais controlada em comparação com os modelos LSTM, o que reflete certa estabilidade em sua capacidade preditiva.

Os modelos LSTM apresentaram um desempenho inconsistente, com altos valores de RMSE e sMAPE, que pioraram à medida que a janela de previsão aumentou. Em especial, o modelo com janela de 40 períodos teve um desempenho notavelmente ruim, sugerindo que os LSTMs têm dificuldade em captar as dinâmicas de longo prazo do preço do Bitcoin. Em contraste, os modelos GRU demonstraram um desempenho superior, especialmente nas janelas de 10 e 20 períodos, com o GRU de 10 períodos se destacando por apresentar o menor RMSE (10.622,67) e o menor sMAPE (14,50%). No entanto, assim como os LSTM, os GRU também sofreram deterioração no desempenho em janelas maiores, especialmente na de 40 períodos.

Dado que o preço do Bitcoin aumentou durante o período de previsão, enquanto a segunda variável preditora permaneceu constante, seria interessante incorporar novos dados que capturam melhor a dinâmica do mercado, como volume de transações e indicadores macroeconômicos. Além disso, testar janelas intermediárias (15 ou 25 períodos) poderia revelar um ponto ideal para o comportamento do Bitcoin, já que o modelo GRU com janela de 10 períodos apresentou o melhor desempenho geral. Uma mudança na frequência dos dados também pode apresentar potencial, uma vez que aumentaria também a amostra.

## **Conclusão**

Ao longo deste trabalho, buscou-se comparar diferentes modelos de previsão de séries temporais aplicados ao preço do Bitcoin, com ênfase em três abordagens: ARIMA-GARCH, LSTM e GRU. A alta volatilidade do Bitcoin e sua relação com fatores externos, como a taxa de juros americana, foram levados em conta nesta pesquisa, justificando a escolha de diferentes metodologias para prever suas variações.

Os resultados indicam que, embora o modelo ARIMA-GARCH tenha mostrado uma boa estabilidade em termos de erro percentual, os modelos baseados em GRU com janelas de previsão de médio prazo apresentaram melhor desempenho preditivo. Entretanto, a performance dos modelos deteriorou-se à medida que se aumentou o horizonte de previsão, sugerindo que o comportamento de longo prazo do Bitcoin é de difícil captura, mesmo por modelos de aprendizado profundo. Dessa forma, conclui-se que, embora os modelos tradicionais como ARIMA-GARCH ainda tenham relevância no contexto de ativos voláteis, as redes neurais recorrentes, especialmente o GRU, demonstram um potencial significativo para previsões de curto prazo. No entanto, é importante destacar que a natureza imprevisível do Bitcoin, aliada a sua alta sensibilidade a fatores exógenos, limita a precisão dos modelos aplicados.

Para pesquisas futuras, recomenda-se a inclusão de variáveis adicionais, como volume de transações, indicadores macroeconômicos ou análises de sentimentos, a fim de aprimorar a capacidade preditiva dos modelos. Além disso, a utilização de janelas intermediárias de previsão e aprofundamento das redes com a inclusão de mais camadas pode ser uma alternativa interessante para capturar dinâmicas temporais mais precisas. De modo geral, os achados alinham-se com o encontrado em pesquisas utilizadas como referências nesse trabalho, apesar do desempenho excepcionalmente ruim da arquitetura LSTM.

## **Referências**



Aboura, S. 2022. A note on the bitcoin and fed funds rate. *Empirical Economics* 63: 2577-2603.

Arch. 2024. ARCH models in Python. Disponível em: <<https://github.com/bashtage/arch>>. Acesso em: 24 mar. 2024.

CoinMarketCap. 2024. Bitcoin price today, BTC to USD live price, marketcap and chart. Disponível em: <<https://coinmarketcap.com/currencies/bitcoin/historical-data/>>. Acesso em: 21 set. 2024.

Bengio, Y.; Simard, P.; Frasconi, P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* 5(2): 157-166.

Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. 2011. Algorithms for hyper-parameter optimization. *25th Annual Conference on Neural Information Processing Systems*: 1-9.

Bergstra, J.; Bengio, Y. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13: 281-305.

Bollerslev, T. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 31: 307-327.

Box, G.E.P.; Jenkins, G.M. 1968. Some Recent Advances in Forecasting and Control. *Journal of the Royal Statistical Society* 17: 91-109.

Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In: *NIPS 2014 Workshop on Deep Learning*, 2014

Dickey, D.A. Fuller, W.A. 1979. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association* 74(366): 427-431.

Dyhrberg, A.H. 2016. Bitcoin, gold and the dollar: A GARCH volatility analysis. *Finance Research Letters* 16: 85-92.

Engle, F.E. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom Inflation. *Econometrica* 50(4): 987-1008.

Federal Reserve Bank of ST Louis [FRED]. 2024. Federal Funds Effective Rate (FEDFUNDS). Disponível em: <<https://fred.stlouisfed.org/series/FEDFUNDS>>. Acesso em: 21 set. 2024.

Ferdiansyah; Othman, S.H.; Radzi, R.Z.R.M.D.; Stiawan, D.; Sazaki, Y.; Ependi, U. 2019. A LSTM-method for bitcoin price prediction: a case study Yahoo finance stock market. *International Conference on Electrical Engineering and Computer Science*: 206-210.

Forbes Digital Assets. 2024. Cryptocurrency prices, market cap and charts. Disponível em: <[Cryptocurrency Prices, Market Cap and Charts | Forbes Digital Assets](#)>. Acesso em: 23 mar. 2024.

Fraz, A.; Hassan, A.; Chughtai, S. 2019. Seasonality in bitcoin market. *NICE Research Journal* 12(1).

Goodwin, P.; Lawton, R. 1999. On the asymmetry of the symmetric MAPE. *International Journal of Forecasting* 15: 405-408.

Haber, S.; Stornetta, W.S. 1991. How to time-stamp a digital document. *Journal of Cryptology* 3: 99-111.

Hamayel, M.J.; Owda, A.Y. 2021. A novel cryptocurrency price prediction model using GRU, LSTM and bi-LSTM machine learning algorithms. *AI* 2:477-496.

Hochreiter, S.; Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8): 1735-1780

Hyndman, R.J.; Koehler, A.B. 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting* 22: 679-688.

Kaiser, L. 2019. Seasonality in cryptocurrencies. *Finance Research Letters* 31: 232-238.

Katsiampa, P. 2017. Volatility estimation for bitcoin: A comparison of GARCH models. *Economics Letters*.

Kayal, P.; Rohilla, P. 2021. Bitcoin in the economics and finance literature: a survey. *SN Business & Economics* 88(1): 1-21.

Keras. 2024. Keras: deep learning for humans. Disponível em: <<https://keras.io/>>. Acesso em: 24 mar. 2024.

Kumar, A.S.; Padakandla, S.R. 2022. Testing the safe-haven properties of gold and bitcoin in the backdrop of COVID-19: A wavelet quantile correlation approach. *Finance Research Letters* 47.

Köse, N.; Ünal, Emre. 2023. The asymmetric effects of the interest rate on the bitcoin price. *Finance Research Letters* 73(2): 189-217.

Lin, Q. 2023. Price prediction of bitcoin using LSTM neural network. *BCP Business & Management* 38: 2738-2744.

Long, H.; Zaremba, A.; Demir, E.; Szczygielski, J.J.; Vasenin, M. 2020. Seasonality in the cross-section of cryptocurrency returns. *Finance Research Letters* 35.

Mohammadi, H.; Su, L. 2010. International evidence on crude oil price dynamics: applications of ARIMA-GARCH models. *Energy Economics* 32: 1001-1008.

Nakamoto, S. 2008. Bitcoin: a peer-to-peer electronic cash system. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 17 mar. 2024.

Narendra, B. C.; Eswara, R. B. 2015. Prediction of selected indian stock using a partitioning-interpolation based ARIMA-GARCH model. *Applied Computing and Informatics* 11: 130-143.

Nedved, M.; Krisfoufek, L. 2023. Safe havens for bitcoin. *Finance Research Letters* 51.

Neves, R.H. 2020. Bitcoin pricing: impact of attractiveness variables. *Financial Innovation* 6(21).

Nguyen, K.Q. 2022. The correlation between the stock market and bitcoin during COVID-19 and other uncertainty periods. *Finance Research Letter* 46.

Patel, M.M.; Tanwar, S.; Gupta, R.; Kumar, N. 2020. A Deep Learning-based cryptocurrency

price prediction scheme for financial institutions. *Journal of Information Security and Applications* 55.

Phillip, A.; Chan, J.; Peiris, S. 2017. A new look at cryptocurrencies. *Economics Letters*.

Seabe, P.L.; Moutsinga, C.R.B.; Pindza, E. 2023. Forecasting cryptocurrency crices using LSTM, GRU, and bi-directional LSTM: a deep learning approach. *Fractal and Fractional* 7(203).

Qezelbash, M.; Tajdini, S.; Jafari, F.; Ghahroudi, M. L.; Farajnezhad, M. 2023. An analysis of volatility and herd behavior among investors in the SP500 stock market index, bitcoin, and gold markets. *Jornal of Mathematics and Modeling in Finance* 3(2): 77-92.

Shanaev, S.; Ghimire, B. 2022. A generalised seasonality test and applications for cryptocurrency and stock market seasonality. *Quarterly Review of Economics and Finance* 86: 172-185.

Shcherbakov, M.; Brebels, A.; Shcherbakova, N.; Tyukov, A.; Janovsky, T.; Kamaev, V. 2013. A survey of forecast error measures. *World Applied Sciences Journal* 24: 171–176.

Shen, Z.; Qing, W.; Leatham, D.J. 2020. Bitcoin return volatility forecasting: a comparative study between GARCH and RNN. *Journal of Risk and Financial Management* 14(7): 337.

Srivastana, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I. Salakhutdinov, R. 2014 Dropout: a simple way to prevent meural networks from overfitting. *Journal of Machine Learning* 15: 1929-1958.

Sudhakaran, P.; Sharma, V.; Khandelwa S. 2024. Bitcoin price prediction using recurrent neural networks and long short-term memory. *E3S Web of Conferences* 491.

Vranken, H. 2017. Sustainability of bitcoin and blockchains. *Current Opinion in Environmental Sustainability* 28: 1-9.

Wang, X.; Chen, X.; Zhao, P. 2020. The relationship between bitcoin and stock market. *International Journal of Operations Research and Information Systems* 11(2): 22-35.