


**ANALISANDO A EFICÁCIA DE ALGORITMOS DE PREVISÃO DO WEKA PARA
A PREVISÃO DO COMPORTAMENTO DO BITCOIN**


**ANALYZING THE EFFECTIVENESS OF WEKA PREDICTION ALGORITHMS
FOR FORECASTING BITCOIN BEHAVIOR**


Recebido em: 16/09/2025

Aceito em: 21/03/2026

Publicado em: 21/04/2026

Daniel Pimenta Gonçalves da Fonte¹ 
Universidade Tecnológica Federal do Paraná

Sandro Carvalho Izidoro² 
Universidade Federal de Itajubá

José Donizetti de Lima³ 
Universidade Tecnológica Federal do Paraná

Sergio Luiz Pessa Ribas⁴ 
Universidade Tecnológica Federal do Paraná

Matheus Henrique Dal Molin Ribeiro⁵ 
Universidade Tecnológica Federal do Paraná

Érick Oliveira Rodrigues⁶ 
Universidade Tecnológica Federal do Paraná

Resumo: Este estudo investigou o desempenho preditivo do software WEKA na previsão de séries temporais do Bitcoin (BTC), comparando seus resultados com aqueles obtidos por meio do R, utilizando dados históricos de maio de 2013 a dezembro de 2024 (4.254 observações). A análise empregou múltiplos algoritmos de aprendizado de máquina avaliados através das medidas MAPE, RMSE e MAE. Os resultados revelaram que, embora o R obteve desempenho superior na maioria dos modelos, especialmente em termos de redução de erro, o WEKA apresentou desempenho competitivo e em alguns casos superior em modelos específicos: M5Rules alcançou MAPE de 2,25% e RMSE de 1.449,49, enquanto SMOreg obteve MAPE de 2,40% e RMSE de 1.450,25 na aba Forecast. Os modelos baseados em regras, M5Rules e M5P, apresentaram desempenho significativamente superior no R, com MAPE de 3,79% e RMSE de 2.112,25, em contraste com os resultados modestos no WEKA, onde apresentaram correlação próxima de zero na aba Classify. O teste KPSS confirmou a não-estacionariedade da série (p-valor = 0,01), e o teste de causalidade de Granger identificou dependência temporal significativa no lag 5 (p-valor = 8,7e-06). Os achados indicam que, embora o WEKA ofereça funcionalidade adequada para modelos específicos em

¹ Aluno do Programa de Pós-graduação PPGEPS da Universidade Tecnológica Federal do Paraná. Brasil, Paraná, Pato Branco. E-mail: danielpimentagdf@gmail.com

² Professor associado à Universidade Federal de Itajubá. Brasil, Minas Gerais, Itabira. E-mail: sandroizidoro@unifei.edu.br

³ Professor do Programa de Pós-graduação PPGEPS da Universidade Tecnológica Federal do Paraná. Brasil, Paraná, Pato Branco. E-mail: donizetti@utfpr.edu.br

⁴ Professor do Programa de Pós-graduação PPGEPS da Universidade Tecnológica Federal do Paraná. Brasil, Paraná, Pato Branco. E-mail: slpessa@professores.utfpr.edu.br

⁵ Professor do Programa de Pós-graduação PPGEPS da Universidade Tecnológica Federal do Paraná. Brasil, Paraná, Pato Branco. E-mail: mribeiro@professores.utfpr.edu.br

⁶ Professor do Programa de Pós-graduação PPGEPS da Universidade Tecnológica Federal do Paraná. Brasil, Paraná, Pato Branco. E-mail: erickrodrigues@professores.utfpr.edu.br

tarefas de previsão sequencial, apresenta instabilidades e limitações na modelagem temporal abrangente, tornando o R uma alternativa mais robusta para aplicações financeiras complexas.

Palavras-chave: BTC; Algoritmos; Séries Temporais; WEKA; Criptomoeda.

Abstract: This study investigated the predictive performance of WEKA software in Bitcoin (BTC) time series forecasting, comparing its results with those obtained through R using historical data from May 2013 to December 2024 (4,254 observations). The analysis employed multiple machine learning algorithms evaluated through MAPE, RMSE, and MAE measures. Results revealed that while R generally achieved superior performance across most models, particularly in terms of error reduction, WEKA showed competitive and in some cases, superior performance in specific models: M5Rules reached MAPE of 2.25% and RMSE of 1,449.49, while SMOreg obtained MAPE of 2.40% and RMSE of 1,450.25 in the Forecast tab. Rule-based models, M5Rules and M5P, exhibited notably stronger performance in R, with MAPE of 3.79% and RMSE of 2,112.25, compared to modest results in WEKA where they showed correlation near zero in the Classify tab. The KPSS test confirmed series non-stationarity (p-value = 0.01), and Granger causality test identified significant temporal dependence at lag 5 (p-value = 8.7e-06). Findings indicate that while WEKA offers adequate functionality for specific models in sequential forecasting tasks, it exhibits instabilities and limitations in comprehensive temporal modeling, making R a more robust alternative for complex financial applications.

Keywords: BTC; Algorithms; Time Series; WEKA; Cryptocurrency.

INTRODUÇÃO

O surgimento das criptomoedas como ativos financeiros digitais introduziu novos paradigmas nos cenários econômico e tecnológico. O Bitcoin, a mais proeminente entre essas moedas digitais, tem atraído crescente atenção de investidores, pesquisadores e analistas devido à sua elevada volatilidade, natureza descentralizada e dinâmicas complexas de preços. Tais características impõem desafios e oportunidades singulares à previsão de séries temporais, um campo que tem evoluído significativamente com a incorporação de técnicas de aprendizado de máquina.

Nos últimos anos, diversas ferramentas foram desenvolvidas para facilitar a aplicação de algoritmos de aprendizado de máquina na análise de séries temporais. Dentre elas, destaca-se o Waikato Environment for Knowledge Analysis (WEKA), um conjunto de softwares de código aberto amplamente utilizado em tarefas de mineração de dados e modelagem preditiva. Embora o WEKA seja frequentemente empregado em problemas de classificação e agrupamento, seu potencial para lidar com tarefas de regressão dependentes do tempo, como a previsão de preços de criptomoedas, ainda é relativamente pouco explorado na literatura acadêmica.

Modelos estatísticos tradicionais, como a regressão linear e os modelos autorregressivos integrados de médias móveis (ARIMA), têm demonstrado eficácia limitada quando aplicados a séries temporais financeiras voláteis, muitas vezes falhando em capturar padrões não lineares e mudanças abruptas no comportamento do mercado. Por outro lado, os modelos de aprendizado de máquina oferecem maior flexibilidade e a capacidade de incorporar diferentes

atributos dos dados. No entanto, a adequação de plataformas específicas, como o WEKA, nesse contexto, exige investigação empírica adicional, especialmente quando comparada a ambientes mais amplamente adotados, como o R.

Este estudo é motivado pela crescente demanda por ferramentas precisas e acessíveis para a previsão de movimentos nos preços de criptomoedas, bem como pela escassez de avaliações abrangentes das capacidades do WEKA nesse domínio. Assim, o principal objetivo deste artigo é avaliar a viabilidade e a eficácia do uso da plataforma WEKA, sendo uma ferramenta mais direta e objetiva para a introdução para o público sem conhecimento específico na área de programação nas áreas de análise de séries temporais e na previsão de preços de criptomoedas, tendo o Bitcoin como estudo de caso. Ao comparar diversos algoritmos de aprendizado de máquina implementados no WEKA com suas versões equivalentes no R, esta pesquisa busca identificar as potencialidades e limitações de cada ambiente na captura de padrões temporais complexos, contribuindo, dessa forma, para o avanço metodológico das abordagens de previsão no contexto das criptomoedas.

Este estudo integra uma coletânea de artigos que investiga a aplicabilidade de algoritmos de aprendizado de máquina na previsão de criptomoedas, com foco inicial no Bitcoin. Os achados aqui discutidos servem como base para os estudos posteriores com altcoins voltadas a jogos blockchain e para análises de transferência de aprendizado entre ativos, apresentados nos artigos subsequentes da coletânea.

TRABALHOS RELACIONADOS

Derbentsev *et al.* (2020) buscaram avaliar a eficiência de dois modelos de aprendizado de máquina Random Forest (RF) e Gradient Boosting Machine (GBM) na previsão dos valores de criptomoedas com um dia de antecedência, com base em seus valores passados e em dois fatores adicionais. O número de defasagens escolhido foi sete, devido à natureza recorrente das transações, que ocorrem sem intervalos. Além da variável de preço diário, os autores incorporaram dois conjuntos de dados adicionais: o volume diário negociado e a taxa de crescimento defasada em um dia desses volumes. As criptomoedas selecionadas para o estudo foram BTC, ETH e XRP, e as medidas de desempenho utilizadas para avaliar a eficácia dos modelos de ensemble foram o Erro Percentual Absoluto Médio (MAPE) e a Raiz do Erro Quadrático Médio (RMSE). Os resultados das previsões fora da amostra indicaram que o modelo GBM de curto prazo teve melhor desempenho na previsão de BTC e XRP, enquanto o RF apresentou melhores resultados para o ETH.



Chowdhury *et al.* (2020) empregaram modelos de aprendizado de máquina, incluindo redes neurais, árvores de Gradient Boosting Machine (GBM), métodos de ensemble e k-Vizinhos Mais Próximos (K-NN), para prever os movimentos de curto prazo dos preços de nove criptomoedas distintas. As principais medidas de avaliação utilizadas foram RMSE, erro absoluto e correlação. As variáveis selecionadas incluíram os preços de abertura e fechamento diários, com os conjuntos de teste e previsão cobrindo os 31 dias de janeiro de 2019. Variáveis adicionais compreenderam os preços máximos e mínimos diários, volumes de transações diárias e a capitalização de mercado de cada criptomoeda. O melhor desempenho foi observado no modelo de ensemble, que alcançou uma acurácia de 92,4%, enquanto o modelo K-NN apresentou o pior desempenho, devido à sua dificuldade em lidar com a extrema volatilidade dos dados.

Haryadi *et al.* (2022) testaram a acurácia do modelo de Máquina de Vetores de Suporte (SVM) na previsão do preço da criptomoeda Polkadot, que, à época, ocupava uma posição de destaque entre as criptomoedas com maior capitalização de mercado global. A variável de entrada selecionada foi o preço de fechamento diário, no período de agosto de 2020 a dezembro de 2021, totalizando 449 pontos de dados. As medidas de avaliação de desempenho incluíram o uso do kernel Gaussiano (RBF), kernel linear e o MAPE. Em termos de acurácia no conjunto de teste, o melhor desempenho foi alcançado com o kernel RBF, que obteve 90% de acurácia e um MAPE de 5,28%.

Samson (2024) avaliou a eficiência de diferentes modelos de aprendizado de máquina na previsão dos preços de fechamento diário das seis maiores criptomoedas disponíveis no mercado à época. O conjunto de dados contemplou períodos variados, abrangendo de 1º de janeiro de 2015 a 26 de janeiro de 2024. Os dados foram divididos em conjuntos de treinamento e teste, na proporção de 80%-20%, respectivamente. As medidas de avaliação utilizadas foram RMSE, R^2 , R^2 ajustado, Erro Quadrático Médio (MSE) e Erro Absoluto Médio (MAE). Os resultados indicaram que o modelo Bagging apresentou o pior desempenho em todas as medidas para todas as criptomoedas analisadas, enquanto o GBM obteve o melhor desempenho apenas para BTC e Solana. Para as demais criptomoedas, o modelo Random Forest superou os demais.

Sun (2024) teve como objetivo avaliar a eficácia de diferentes modelos na previsão de preços, utilizando XGBoost, LightGBM e Redes Neurais Bayesianas (BNN). O estudo empregou busca em grade (grid search) e validação cruzada para avaliar o desempenho dos parâmetros selecionados em diferentes subconjuntos. Entre os parâmetros mais comuns analisados estavam a taxa de aprendizado (learning rate), a profundidade máxima das árvores e

outros. As medidas de desempenho para o conjunto de treinamento incluíram RMSE, MSE, MAE e MAPE. Os resultados das previsões demonstraram que o modelo XGBoost superou significativamente os outros dois modelos em todas as criptomoedas, alcançando uma taxa de erro percentual de 3,5%. A criptomoeda com melhor desempenho foi a LTC; contudo, na medida MAPE do modelo BNN, a LTC apresentou o pior resultado entre as três analisadas, com uma taxa de erro de 73,27%, em comparação com 21,39% para BTC e 12,49% para ETH.

Tabela 1 – Estudos relacionados.

Autores	Criptomoedas	Modelos
Yenidogan <i>et al.</i> (2018)	BTC	PROPHET, ARIMA
Derbentsev <i>et al.</i> (2019)	BTC, ETH, XRP	BART, ARIMA, ARFIMA
Derbentsev <i>et al.</i> (2020)	BTC, ETH, XRP	GBM, RF
Derbentsev <i>et al.</i> (2020)	BTC, ETH	SVM, ANN, RF, GBM
Chowdhury <i>et al.</i> (2020)	cci30	Neural Networks, GBM tree, ensemble, K-NN
Haryadi <i>et al.</i> (2022)	Polkadot	RBF, Kernel Linear
Kurniawan <i>et al.</i> (2022)	BTC, XRP, LTC	ARIMA, GARCH, Holt-Winters
Contreras <i>et al.</i> (2024)	BTC	SVM, RF, MLP, KNN, LR, Decision Tree
Samson (2024)	BTC, ETH, SOL	Bagging, GBM, RF
Sun (2024)	BTC, ETH, LTC	XGB, LGBM, BNN

Fonte: Elaborado pelos autores (2025).

PROCESSOS METODOLÓGICOS

Este estudo foi conduzido com base em dados históricos do Bitcoin (BTC), abrangendo o período de 3 de maio de 2013 a 26 de dezembro de 2024, totalizando 4.254 observações extraídas da base pública da Coingecko. O conjunto de dados original, denominado “btc-usd-max.csv”, continha dez atributos principais: data (snapped_at), preço de fechamento (price), capitalização de mercado (market_cap), volume diário de transações (total_volume), cinco defasagens temporais do preço (lags de 1 a 5), a média móvel de cinco períodos do preço e o valor atual utilizado como variável alvo.

O pré-processamento foi realizado por meio de um script em Python, que leu os valores numéricos do arquivo CSV original, construiu uma janela temporal de cinco observações passadas para cada instância, calculou a média das observações, e adicionou o valor atual do preço como variável alvo. O arquivo resultante, “btc-usd-windowed.csv”, foi salvo com os



nomes dos atributos padronizados, permitindo seu uso direto na plataforma WEKA para análise e modelagem preditiva.

VALIDAÇÃO ESTATÍSTICA E ANÁLISE EXPLORATÓRIA

Antes da aplicação dos modelos, foi conduzida uma análise estatística preliminar no ambiente R, com o objetivo de validar a consistência das variáveis e sua adequação à modelagem preditiva. Inicialmente, foi aplicada a função “complete.cases()” para remover eventuais observações com valores ausentes, assegurando a integridade dos dados.

O teste de estacionariedade KPSS foi aplicado à variável price, utilizando uma diferenciação com 10 lags. O valor-p obtido rejeitou a hipótese nula de estacionariedade, caracterizando a série como não estacionária. Em seguida, aplicou-se o teste de causalidade de Granger para avaliar a dependência entre volume e price, com cinco defasagens. Os resultados indicaram que o volume influencia significativamente o preço (valor-p = 0,04947), e que a relação inversa também é verdadeira (valor-p = 0,01105).

Ainda que ambas as variáveis apresentem causalidade mútua, os resultados apontaram que o preço exerce influência mais expressiva sobre o volume. Essa conclusão foi reforçada por um modelo VAR (Vetores Autorregressivos) com três diferenciações, no qual os preços defasados apresentaram valores-p de 0,0134 (lag 1) e 0,0438 (lag 2), ambos estatisticamente significativos. Para o volume, a correlação residual de 0,06997 indicou uma influência menos pronunciada.

Outros testes complementares foram realizados, como o teste de tendência de Cox-Stuart, cujos valores-p inferiores a $2,2e-16$ confirmaram que as séries de preço e volume não são aleatórias. Já o teste de Kruskal-Wallis, utilizado para verificar diferenças entre grupos temporais, indicou ausência de variação significativa para o preço (valor-p = 0,1274), mas forte significância estatística para o volume (valor-p = $6,078e-07$), sugerindo alterações estruturais na distribuição do volume ao longo do tempo.

MODELAGEM PREDITIVA E VALIDAÇÃO

O conjunto de dados final, contendo as variáveis defasadas e agregadas, foi dividido em 70% para treinamento e 30% para teste, respeitando a ordem temporal (validação fora da amostra). Esse procedimento visou garantir a avaliação realista dos modelos com base em dados que não foram expostos durante o aprendizado.

A modelagem preditiva foi realizada em R e WEKA, utilizando uma variedade de algoritmos que pudesse conter no repositório deste último programa. Os modelos incluíram



Regressão Linear (LR), Random Forest (RF), Gaussian Processes, M5Rules, M5P, Multilayer Perceptron (MLP), SMOreg, Alternating Model Tree, Decision Stump, Bagging, Stacking e IBk.

Todos os algoritmos foram avaliados com base nas seguintes medidas de desempenho padronizadas: i) Erro absoluto médio (MAE); ii) Raiz do erro quadrático médio (RMSE); iii) Erro percentual absoluto médio (MAPE); iv) Correlação de Pearson (R).

Complementarmente, foi realizada uma modelagem específica com horizonte de um dia à frente, utilizando a funcionalidade de previsão (forecast) do WEKA, com os mesmos atributos e critérios de validação.

ALGORITMOS UTILIZADOS PELO WEKA

A seguir, uma demonstração dos algoritmos presentes no WEKA bem como uma descrição de seu funcionamento e da configuração utilizada para mensurar os resultados de cada teste presente.

LINEAR REGRESSION

Segundo Su *et al.* (2012), a regressão linear fornece o modelo mais simples entre as opções disponíveis para realizar regressão como um combinador preditivo linear. Isso se deve à sua estrutura, que tende a ser mais eficaz quando o tamanho da amostra é pequeno ou quando os dados apresentam correlações fracas. Nesses casos, a regressão linear tende a produzir resultados eficazes.

$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \varepsilon_i, \text{ com } \varepsilon_i \stackrel{iid}{\sim} N(0, \sigma^2) \quad (1)$$

Onde y_i é a i -ésima resposta medida em escala contínua, x_i é o vetor de preditores associado, p será o número total de variáveis independentes e i um valor variável de 1 até N , sendo esse último o tamanho da amostra. Os coeficientes β_n definem a estrutura linear do modelo, sendo β_0 o intercepto, que indica o valor de y_i quando todos os preditores apresentam valores nulos. A variável ε_i é o termo de erro aleatório que captura a variação não explicada pelo modelo, assumido como independente e identicamente distribuído segundo uma distribuição normal de média zero e variância σ^2 sendo o pressuposto fundamental para a validade das inferências

realizadas. A configuração utilizada no WEKA baseou-se no atributo “M5 Method” com 100 instâncias e um parâmetro ridge definido como 1.0E-8.

GAUSSIAN PROCESSES

Segundo Yan *et al.* (2015), o modelo de Processos Gaussianos realiza previsões a partir de uma distribuição conjunta sobre os dados, derivada de uma função de covariância que quantifica a similaridade entre pares de observações. A predição para um novo ponto é obtida como a média da distribuição condicional posterior, dada a observação dos dados de treinamento. Os hiperparâmetros do modelo são estimados de forma a maximizar a verossimilhança marginal dos dados. No presente estudo, o número de instâncias foi 100, o tipo de filtro foi definido como dados de treinamento normalizados, com ruído = 1,0, e o kernel selecionado foi o PolyKernel.

MLP

Segundo Ramchoun *et al.* (2016), o Multilayer Perceptron (MLP) é a forma mais convencional de rede neural, utilizando o algoritmo de retropropagação como método de treinamento. Um aspecto importante é a configuração da estrutura do modelo, pois a escolha ideal depende do número de camadas, número de nós ocultos em cada camada e da função objetivo. Segundo Ali *et al.* (2020), o MLP mapeia os dados de entrada para os de saída ajustando os pesos dos nós internos. A função de aprendizado do modelo, dada na retropropagação, é expressa como: $f(x) = R^i > R^0$. A configuração envolveu uma taxa de aprendizado de 0.3 e um valor de momento de 0.2, aplicado às atualizações de peso. O parâmetro das camadas ocultas foi definido como “a”, e o número de épocas de treinamento foi 500.

RBF NETWORKS

Segundo Zhang e Zhang (2004), a rede RBF possui forte capacidade de mapeamento não linear e alta velocidade de treinamento em comparação com outros modelos de redes neurais. A rede é composta por três camadas do tipo feedforward sendo entrada, oculta e saída, em que cada unidade da camada oculta aplica uma função de base radial centrada em um protótipo aprendido durante o treinamento, e a saída é obtida como uma combinação linear ponderada das ativações dessas unidades. A configuração escolhida no WEKA incluiu tamanho de lote de 100, semente de clusterização igual a 1, número máximo de iterações da regressão

logística igual a -1, desvio padrão mínimo entre clusters de 0,1, dois clusters e valor ridge de 1,0E-8.

SMOREG

Segundo Reddy *et al.* (2021), o algoritmo em questão realiza tarefas de classificação multiclasse e problemas de regressão, convertendo valores nominais em valores numéricos com base nos dados de entrada. De acordo com Poyraz e Gürhanlı (2020), o algoritmo opera aplicando regressão em modelos de SVM. Já Singh *et al.* (2013) afirmam que esse algoritmo é uma evolução do SMO básico, sendo particularmente utilizado para previsões não lineares. O tamanho do lote foi definido como 100, o parâmetro de complexidade C foi configurado como 1.0, o tipo de filtro foi “normalized training data” com PolyKernel, e o algoritmo de aprendizado utilizado foi o RegSMOImproved.

IBK

Segundo Ali *et al.* (2020), o algoritmo serve para tarefas de classificação e regressão. Sendo um modelo pertencente ao grupo de aprendizado preguiçoso (lazy learning), ou seja, um dos modelos K-NN, sua utilização depende da distância entre dois vetores de dados, utilizando a distância Euclidiana para dados contínuos e a distância de Hamming para dados discretos. O número de vizinhos usado no WEKA foi 1, com 100 instâncias. Não foi aplicada validação cruzada nem ponderação por distância. O algoritmo escolhido foi o LinearNNSearch.

KSTAR

Segundo Sabanci *et al.* (2016), o KStar é um classificador baseado em distância, sendo sua principal diferença em relação a outros métodos desse tipo a função de distância utilizada, que, nesse caso, é baseada em entropia. A configuração incluiu 100 instâncias, blending global definido como 20, e o modo de tratamento de valores ausentes foi definido como média das curvas de entropia por coluna.

ADDITIVE REGRESSION

Segundo Horowitz (2006), o modelo de Regressão Aditiva é representado conforme a equação abaixo:

$$Y^i = m_0 + m_1(X_i^1) + \dots + m_D(X_i^D) + \varepsilon_i, \quad i = 1, \dots, n \quad (2)$$

Sendo Y^i as variáveis resposta, e as covariáveis $X^i = (X_i^1 \dots X_i^D)$ termos aleatórios identificados como independentes e identicamente distribuídos. A constante m_0 e as funções (m_1, \dots, m_D) são valores desconhecidos, ε_i é o erro aleatório associado à i -ésima observação e n o tamanho da amostra. Foram utilizadas 100 instâncias, o classificador foi o Decision Stump, o número de iterações foi 10 e o parâmetro de shrinkage foi 1.0.

ATTRIBUTE SELECTED CLASSIFIER

Segundo Gnanambal (2018), o modelo é baseado em duas etapas: a primeira é responsável por gerar o subconjunto de atributos, e a segunda por ordená-lo. A geração do subconjunto depende de um processo de busca que compara o novo subconjunto gerado com o anterior. Esse processo se repete até que o subconjunto atinja a condição de parada. A ordenação, por sua vez, define os valores dos atributos, o que pode ser feito estatisticamente ou com base em informações teóricas. Com 100 instâncias, o classificador utilizado foi o Random Tree, o avaliador de atributos foi o cfsSubsetEval e o método de busca aplicado foi o BestFirst.

BAGGING

Segundo Niranjan (2018), o Bagging é uma das técnicas de ensemble, envolvendo o “sorteio” de t instâncias aleatórias para um conjunto de treinamento de tamanho t , utilizando uma distribuição uniforme e aprendendo com essas t instâncias. Trata-se de um processo repetitivo que continua a ser executado diversas vezes e a configuração consistiu em 100 instâncias, com o classificador REPTree e 10 iterações.

CVPARAMETERSELECTION

A seleção dos hiperparâmetros do modelo foi realizada por meio do método CVParameter Selection, proposto por Kohavi (1995), que utiliza validação cruzada para identificar os valores ótimos dos parâmetros do classificador base. O método avalia sistematicamente combinações de parâmetros dentro de intervalos predefinidos, selecionando aquela que minimiza o erro de generalização estimado.

Os parâmetros submetidos à busca por validação cruzada foram configurados da seguinte forma: o número de dobras (numFolds) define a quantidade de partições utilizadas na validação cruzada; o parâmetro seed controla a semente do gerador de números aleatórios, garantindo a reprodutibilidade dos resultados; e o parâmetro CVParameters especifica o espaço de busca de cada hiperparâmetro do classificador base

ENSEMBLE SELECTION

A combinação dos modelos foi realizada por meio do método Ensemble Selection, proposto por Caruana *et al.* (2004), que constrói um ensemble otimizado a partir de uma biblioteca de modelos previamente treinados, selecionando iterativamente aqueles que maximizam o desempenho coletivo. O método opera por meio de um processo de escalada de colina (hillclimbing), avaliando em cada iteração quais modelos da biblioteca contribuem para a redução do erro do ensemble.

Os principais parâmetros do método foram configurados da seguinte forma: numFolds define o número de partições utilizadas na validação cruzada; validationRatio determina a proporção dos dados de treinamento reservada para validação interna do ensemble; modelRatio controla a fração dos modelos da biblioteca selecionados aleatoriamente a cada iteração; sortInitializationRatio define a proporção de modelos utilizados na inicialização por ordenação, e o parâmetro greedySortInitialization indica se essa inicialização interrompe a adição de modelos quando o desempenho começa a degradar. O parâmetro replacement determina se um mesmo modelo pode ser incluído mais de uma vez no ensemble, e numModelBags especifica o número de agrupamentos de modelos (model bags) utilizados pelo algoritmo. A medida de otimização do ensemble foi definida pelo parâmetro hillclimbMetric, e o número de iterações do processo de escalada de colina foi controlado por hillclimbIterations. Por fim, seed garante a reprodutibilidade dos resultados por meio do controle da semente do gerador de números aleatórios.

RANDOM COMMITTEE

Segundo Niranjana (2018), esse método envolve a construção de uma base de classificadores utilizando um conjunto de valores de semente aleatórios únicos. O resultado final da classificação é obtido pela média das previsões geradas pelos classificadores base individuais.

O Random Committee foi configurado com RandomTree como classificador base, parametrizado com $K = 0$ (número de atributos selecionados aleatoriamente por divisão, onde 0 indica o uso da raiz quadrada do total de atributos), $M = 1,0$ (peso mínimo total em uma folha) e $V = 0,001$ (variância mínima proporcional para que uma divisão seja realizada). O ensemble foi composto por 10 iterações (numIterations = 10), com batchSize de 100 instâncias e semente inicial (seed) igual a 1, a partir da qual sementes distintas são derivadas para cada árvore individual, garantindo diversidade entre os modelos base. O resultado final da previsão é obtido pela média das previsões numéricas de todas as árvores geradas.

STACKING

Segundo Niranjana (2018), diferentemente do Bagging, esse classificador treina um algoritmo capaz de combinar as previsões de diferentes algoritmos de aprendizado. Cada algoritmo é treinado individualmente com os dados disponíveis, e, por fim, o algoritmo combinador é treinado utilizando as previsões dos outros algoritmos como entradas adicionais antes de fazer a previsão final. Foram usados para esse estudo uma combinação dos classificadores Linear Regression, M5Rules e SMOreg

WEIGHTED INSTANCES HANDLER WRAPPED

Segundo Kargar *et al.* (2021), a principal vantagem desse modelo reside no treinamento das instâncias atribuindo pesos. Caso o classificador desejado não esteja disponível no software utilizado, aplica-se uma técnica de reamostragem de outro algoritmo para atribuir pesos aos dados de treinamento. O uso de aprendizado de máquina meta ou algoritmos base torna o modelo mais robusto. O classificador usado foi o Conjunctive Rule.

DECISION TABLE

Segundo Kohavi (1995), a Decision Table é um classificador que seleciona um subconjunto ótimo de atributos para compor a tabela de decisão, utilizando a maioria das instâncias que compartilham os mesmos valores de atributos para realizar a predição. A seleção dos atributos é conduzida por um algoritmo de busca, que avalia combinações de atributos segundo uma métrica de desempenho predefinida.

A configuração do modelo adotou o algoritmo de busca BestFirst com direção de busca descendente (-D 1) e máximo de cinco nós sem melhora antes da interrupção (-N 5), explorando o espaço de subconjuntos de atributos de forma eficiente. A medida de avaliação utilizada foi o RMSE, adequada ao contexto de regressão do presente estudo. A validação do modelo foi conduzida pelo método leave-one-out, correspondente à validação cruzada com número de dobras igual ao número de instâncias, o que maximiza o uso dos dados disponíveis para estimação do erro de generalização.

M5RULES

Segundo Poyraz e Gürhanlı (2020), o algoritmo utiliza modelos de regressão ou árvores de regressão, empregando uma lista para produzir previsões numéricas. Em árvores de regressão, as folhas representam a média das amostras em cada nó terminal, enquanto neste modelo, as folhas são expressas como equações de regressão linear.

Segundo Fang *et al.* (2020), o modelo M5 gera a melhor regra para cada padrão dentro de cada árvore, gerando árvores completas em vez de subárvores menores. O algoritmo foi configurado com 100 instâncias, e o número mínimo de instâncias para permitir um nó folha foi definido como 4.0.

ZEROR

O ZeroR é descrito pelo WEKA como uma classe para construção e uso de um classificador 0-R, que prediz a média para variáveis numéricas ou a moda para variáveis nominais (FRANK; HALL; WITTEN, 2016). Por ignorar completamente os atributos preditores e basear-se exclusivamente na variável-alvo, sua capacidade preditiva é intrinsecamente limitada. Ainda assim, é amplamente utilizado como baseline, servindo de referência mínima para avaliar o desempenho de classificadores mais complexos.

HOLT-WINTERS

Segundo Kalekar (2004), o método Holt-Winters multiplicativo baseia-se em três equações de atualização recursiva, correspondentes aos componentes de nível, tendência e sazonalidade da série. O nível é estimado pela média ponderada entre a observação atual dessazonalizada e a projeção do período anterior; a tendência é atualizada a partir da variação entre os níveis consecutivos; e o componente sazonal é corrigido pela razão entre o valor observado e o nível estimado. A previsão final é obtida pelo produto entre a projeção linear do nível com a tendência e o índice sazonal correspondente ao horizonte desejado. O algoritmo foi configurado com ciclo sazonal de 12, fator de suavização sazonal de 0,2, fator de suavização da tendência de 0,2 e fator de suavização do valor de 0,2.

RANDOM FOREST

Segundo Breiman (2001), o Random Forest para regressão é formado pelo crescimento de árvores dependentes de um vetor aleatório θ , de modo que cada preditor individual $h(\mathbf{x}, \theta_k)$ assume valores numéricos. A predição final do modelo corresponde à média dos preditores individuais sobre as K árvores da floresta, $av_k h(x, \theta_k)$ cujo erro quadrático médio de generalização é definido como $E_{X,Y}(Y - av_k h(X, \theta_k))^2$, em que X representa os dados de entrada, Y é a variável resposta numérica, θ_k é um vetor aleatório independente e identicamente distribuído que governa o crescimento da k -ésima árvore, av_k denota a média sobre todas as K árvores da floresta. Foram utilizadas 100 instâncias e 100 árvores na floresta.

ALTERNATING MODEL TREE

Segundo Frank *et al.* (2015), árvores de modelo alternante são modelos baseados em árvores de decisão que utilizam boosting e são aplicadas principalmente em problemas de classificação. A construção do modelo distingue dois tipos de nós: nós divisores e nós de previsão. O primeiro corresponde a um nó interno, enquanto o segundo contém os resultados numéricos usados nas previsões. Foram utilizadas 100 instâncias, com 10 iterações e shrinkage configurado como 1.0.

DECISION STUMP

O Decision Stump é uma árvore de decisão de nível único, utilizada geralmente em conjunto com algoritmos de boosting, suportando tanto regressão, baseada no erro quadrático médio, quanto classificação, baseada em entropia, tratando valores ausentes como uma categoria separada (FRANK; HALL; WITTEN, 2016). Docharkhehsaz *et al.* (2022) descrevem que, assim como os demais modelos em árvore, o Decision Stump é baseado em algoritmos de divisão e conquista, nos quais o problema é decomposto em subproblemas menores e mais tratáveis. Sua estrutura é composta por um único nó raiz, que representa a condição ou questão a ser decidida, e ramificações que conduzem diretamente ao resultado da classificação, conforme a condição satisfeita. Os autores destacam ainda que o algoritmo apresenta bom desempenho em dados coletados naturalmente, sendo amplamente utilizado em sistemas computacionais de tempo real que demandam decisões de alta velocidade, com aplicações em processamento de imagens, sistemas de segurança e detecção de ataques cibernéticos. Por outro lado, dados manipulados artificialmente ou estruturados para fins de simplificação do problema podem tornar o algoritmo suscetível a erros

M5P

Segundo Behnood *et al.* (2017), o M5P é um algoritmo derivado do M5, com uma de suas principais vantagens sendo a capacidade de lidar eficientemente com grandes conjuntos de dados, não apenas com muitos atributos, mas também com dados de alta dimensionalidade, tornando-o eficiente e robusto no tratamento de valores ausentes. Além disso, Reddy *et al.* (2021) observam que o modelo M5P constrói árvores de regressão cujas folhas contêm modelos lineares.

Essas árvores são formadas com base na divisão recursiva do espaço de atributos, e as equações lineares são utilizadas nos nós folhas para realizar previsões numéricas. Segundo o

autor, o algoritmo equilibra a complexidade do modelo com a capacidade preditiva ao aplicar podas baseadas na redução do erro. No WEKA, o modelo foi configurado com 100 instâncias, número mínimo de instâncias por folha de 4, uso de podas ativado e suavização aplicada às previsões.

A suavização é um procedimento pós-poda aplicado aos modelos lineares das folhas com o objetivo de evitar descontinuidades nas fronteiras entre regiões da árvore, os coeficientes das equações lineares de folhas adjacentes são combinados de forma ponderada ao longo do caminho da raiz até a folha. Isso resulta em previsões mais suaves nas transições entre regiões, conforme descrito em Quinlan (1992) e Reddy *et al.* (2021).

RANDOM TREE

Segundo Niranjana (2018), este modelo é uma árvore de decisão construída com base em um subconjunto aleatório de atributos. É importante destacar que uma árvore de decisão é composta por nós, ramos e folhas. Cada nó representa um teste sobre um atributo, cada ramo corresponde a uma saída possível, e as folhas representam as previsões finais após a avaliação de todos os atributos na forma de rótulos de classe. Além disso, Kargar *et al.* (2021) afirmam que a Random Tree (RT) é um método rápido e flexível, podendo ser aplicada em diferentes domínios e problemas de otimização variados. A configuração utilizada incluiu: $kvalue = 0$, peso total mínimo em uma folha = 1.0, e variação mínima proporcional = 0.001.

REP TREE

Segundo Paudel e Bhatta (2022), a REP Tree é uma técnica rápida de aprendizado e classificação baseada na arquitetura de árvores de decisão. Complementarmente, conforme descrito pelo próprio WEKA, o algoritmo constrói árvores de decisão ou regressão utilizando ganho de informação ou variância como critério de divisão, aplicando poda por redução de erro com backfitting para controle da complexidade do modelo. Uma característica relevante de eficiência é que os valores dos atributos numéricos são ordenados apenas uma vez durante o processo de construção. Os valores ausentes são tratados por meio da divisão proporcional das instâncias correspondentes entre os ramos, de forma análoga ao procedimento adotado pelo algoritmo C4.5.

O parâmetro `minVarianceProp` define a proporção mínima da variância total dos dados que deve estar presente em um nó para que uma divisão seja realizada em árvores de regressão, controlando assim a granularidade do modelo. O parâmetro `numFolds` determina a proporção dos dados destinada à poda, sendo uma partição reservada para esse fim e o restante utilizado

para o crescimento da árvore. O parâmetro `maxDepth` limita a profundidade máxima da árvore, e `minNum` estabelece o peso mínimo total de instâncias exigido em um nó folha para que ele seja mantido. A configuração adotada neste estudo definiu contagem inicial igual a 0, profundidade máxima igual a -1 (sem restrição), peso total mínimo em uma folha igual a 2,0, variação mínima proporcional igual a 0,001 e número de partições (folds) igual a 3.

RESULTADOS

O teste KPSS (Kwiatkowski–Phillips–Schmidt–Shin) para estacionariedade em nível apresentou um valor estatístico de 25,989 com p-valor de 0,01. Este resultado leva à rejeição da hipótese nula de estacionariedade, sugerindo que a série temporal dos preços do BTC é não estacionária em nível, o que confirma a presença de uma tendência estocástica nos dados. Este achado é crítico, pois a não estacionariedade pode comprometer os pressupostos de diversos modelos de séries temporais caso não seja adequadamente tratada.

O teste de Kruskal-Wallis, uma abordagem não paramétrica para comparação de medianas entre múltiplos grupos, resultou em uma estatística qui-quadrado de 8,0295 com 3 graus de liberdade e um p-valor de 0,04541. Como o p-valor é inferior a 0,05, infere-se que há diferenças estatisticamente significativas entre pelo menos um par de grupos dentro da série de preços. Esse resultado corrobora a hipótese de que a série apresenta heterogeneidade estrutural ao longo dos períodos ou agrupamentos analisados.

O teste de tendência de Cox-Stuart produziu uma estatística de teste de 2127 com p-valor inferior a $2,2e-16$. Esse valor extremamente baixo rejeita fortemente a hipótese nula de aleatoriedade, fornecendo evidências estatísticas robustas de uma tendência sistemática na série de preços do BTC. Esse achado está em conformidade com o comportamento esperado de séries financeiras, nas quais tendências de longo prazo são frequentemente observadas devido a fatores macroeconômicos e de sentimento de mercado.

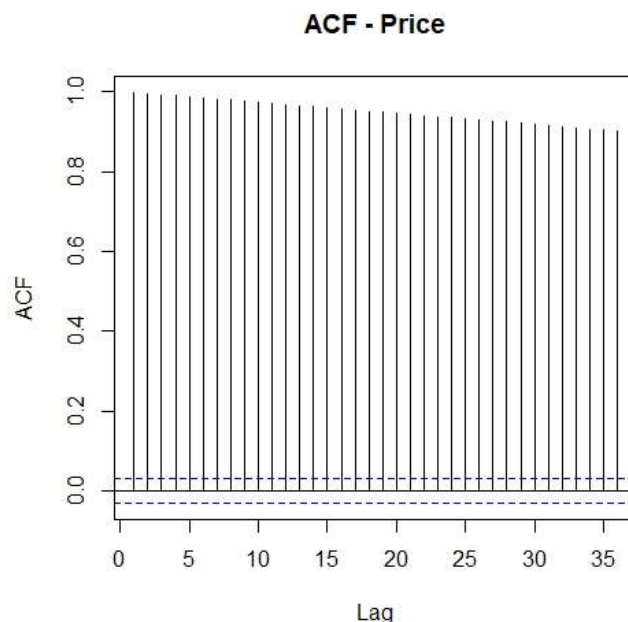
Testes de causalidade de Granger foram aplicados para avaliar se valores passados (lags) dos preços do BTC contêm informações preditivas sobre seus valores futuros. O atraso 1 (Past 1) apresentou uma estatística F, sendo a medida da razão entre a variância explicada pelo modelo e a variância residual, de 3,1045 com p-valor de 0,07815. Embora esse resultado não seja estatisticamente significativo ao nível de 5%, indica uma influência marginal e possível autocorrelação de curto prazo.

Os lags 2 e 3 apresentaram p-valores de 0,6175 e 0,1549, respectivamente, sugerindo ausência de causalidade de Granger significativa nos níveis convencionais. Já o lag 4 apresentou

um p-valor de 0,07182, também levemente acima do limiar de 5%, o que pode refletir padrões de autocorrelação de médio prazo. Destaca-se que o lag 5 (Past 5) obteve uma estatística F de 6,2492 com p-valor altamente significativo de $8,7e-06$. Esse resultado confirma que o quinto retardo do preço do BTC é um preditor estatisticamente significativo de seu valor atual, evidenciando a presença de memória de longo prazo, isto é, observações passadas distantes ainda possuem uma influência estatisticamente significativa sobre os valores presentes sugerindo que a série não seria um processo de ruído branco.

A Figura 1 apresenta a Função de Autocorrelação (ACF) da série de preços do BTC. Observa-se que os coeficientes de autocorrelação decaem lentamente, sem cruzar a banda de confiança em defasagens curtas, o que é característico de séries não estacionárias com tendência.

Figura 1 – Gráfico de teste de autocorrelação.

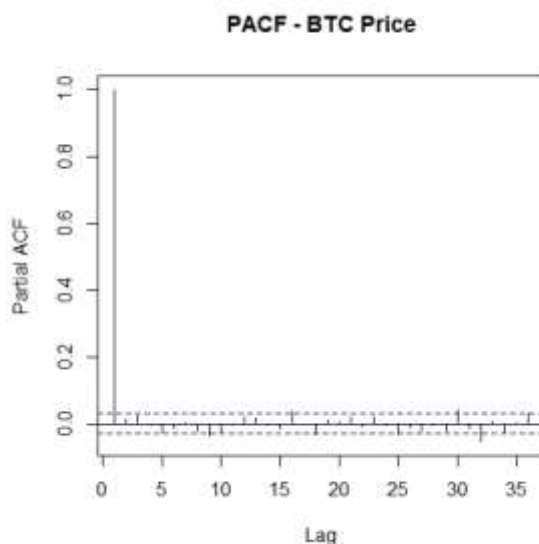


Fonte: elaborado pelos autores com o uso do R (2025).

Em síntese, a combinação dos testes estatísticos aplicados revela que a série temporal de preços do BTC é não estacionária, apresenta tendência e evidencia dependência temporal, especialmente em defasagens mais longas (como a de ordem 5). Essas características devem ser cuidadosamente consideradas durante a modelagem e a previsão dos preços, especialmente na escolha adequada da estrutura de defasagens e na aplicação de transformações que assegurem a estacionariedade dos dados.

A Figura 2 apresenta a Função de Autocorrelação Parcial (PACF), cujo coeficiente elevado na defasagem 1 e rápido decaimento subsequente sugere que um modelo AR(1) pode capturar parcialmente a estrutura de dependência, embora a não-estacionariedade exija transformações previamente à modelagem.

Figura 2 – Gráfico de teste de autocorrelação parcial.



Fonte: elaborado pelos autores com o uso do R (2025).

O coeficiente parcial na defasagem 1 aproxima-se de 1, indicando forte dependência entre cada observação e seu valor imediatamente anterior. A partir da defasagem 2, os coeficientes tornam-se estatisticamente não significativos, mantendo-se dentro do intervalo de confiança representado pelas linhas tracejadas, o que evidencia ausência de dependência direta entre observações mais distantes no tempo. Esse padrão, combinado com o comportamento observado na ACF, é característico de séries não estacionárias com componente autorregressivo de primeira ordem, reforçando a necessidade de diferenciação da série antes da etapa de modelagem.

Apesar de alguns algoritmos terem alcançado seu melhor desempenho possível com os dados fornecidos, a performance geral permaneceu insuficiente para justificar sua aplicação prática em contextos preditivos.

Além disso, observou-se que determinados algoritmos distintos apresentaram desempenhos idênticos. Tal ocorrência decorre dos parâmetros de ajuste selecionados, os quais representavam o resultado ótimo possível para o conjunto de dados de entrada utilizado, composto por 1278 observações.

Alguns modelos apresentaram resultados idênticos, como é o caso dos algoritmos WeightedInstancesHandlerWrapped, Misc. InputMappedClassifier e Decision Table. Essa equivalência pode ser atribuída ao fato de compartilharem a mesma configuração interna, que combina o uso do Decision Table com o método de busca Best First como mecanismo de ajuste.

Tabela 3 – Resultados dos modelos na aba classify.

Algoritmo	Correlação	RMSE	MAE
Additive Regression	0.8360	10799.19	7797.23
AMT	0.9992	1907.89	1438.30
Decision Table	0.9329	8293.91	4246.38
Gaussian Processes	0.9959	3533.31	2567.71
Holt-Winters	0.0000	21893.37	16514.21
IBk	0.9377	7623.99	3641.52
Linear Regression	0.5723	33967.92	28809.24
M5Rules	0.0000	40570.29	35621.42
RBF Networks	0.2895	18838.36	15359.91
Random Committee	0.8160	24209.92	20136.52
Random Forest	0.8086	21103.25	16426.70
Random Tree	0.9094	8233.83	5536.94
REP Tree	0.0000	21669.36	17374.11
SMOreg	0.9985	1239.02	897.68

Fonte: elaborado pelos autores (2025).

O algoritmo Multilayer Perceptron não produziu resultados na aba Classify, mesmo após longos períodos de processamento contínuo. Em algumas ocasiões, chegou a ocasionar falhas no funcionamento do programa WEKA, em razão do consumo excessivo de memória. Ademais, algoritmos como M5Rules e M5P apresentaram resultados idênticos em todas as medidas de desempenho, o que sugere uma possível limitação na forma como esses modelos interpretam o conjunto de dados utilizado para os testes.

Observou-se que os resultados obtidos na aba Classify não foram sistematicamente inferiores aos da aba Forecast, o que indica que a superioridade de desempenho entre as duas abordagens varia conforme o modelo utilizado, sem um padrão consistente. Isso pode ser exemplificado pelos modelos baseados em árvores de decisão: enquanto Random Forest, Random Committee e REP Tree obtiveram desempenho superior na aba Forecast, os algoritmos Random Tree e Alternating Model Tree apresentaram melhores resultados na aba Classify.

Os modelos que apresentaram os resultados mais promissores, combinando altos coeficientes de correlação na aba Classify com baixas taxas de erro na aba Forecast, foram, em

ordem: SMOreg, Alternating Model Tree e Gaussian Processes. De modo geral, o algoritmo SMOreg mostrou-se o mais adequado para fins de previsão neste contexto, sobretudo porque a aba Forecast está melhor alinhada com a natureza temporal dos dados analisados.

Tabela 4 – Resultados dos modelos na aba forecasting.

Algoritmo	MAPE (W)	MAPE (R)	RMSE (W)	RMSE (R)	MAE (W)	MAE (R)
Alternating Tree	38.14%	3.79%	42824.4	2112.2	23711.1	1678.9
Bagging	19.79%	5.55%	9623.7	6884.9	6453.1	2863.7
Decision Stump	45.11%	42.38%	18305.0	17665.3	15369.2	14813.3
Gaussian Processes	7.43%	80.16%	2636.2	40255.0	2274.3	35226.0
IBK	18.82%	7.27%	10367.1	7623.9	7449.3	3641.5
Linear Regression	2.92%	3.778%	1583.9	2107.8	1161.2	1673.2
M5P	3.13%	3.79%	2068.8	2112.2	1412.7	1678.9
M5Rules	2.25%	3.79%	1449.4	2112.2	963.6	1678.9
MLP	12.11%	3.44%	10675.8	2320.5	6615.0	1480.0
Random Forest	16.81%	6.01%	9904.8	7446.5	6500.9	3190.1
REPTree	21.76%	5.82%	9759.7	6969.0	6718.2	2983.5
SMOreg	2.40%	9.89%	1450.2	14490.0	1011.2	6330.6
Stacking	2.83%	80.44%	1528.0	40273.3	1131.7	35282.8

Fonte: elaborado pelos autores (2025).

A Tabela 4 apresenta uma análise comparativa de desempenho entre modelos treinados e avaliados nas plataformas WEKA (W) e R (R), considerando diversas medidas de erro: MAPE, RMSE e MAE. De modo geral, os resultados indicam discrepâncias significativas entre as duas plataformas, com a linguagem R demonstrando desempenho superior em diversos modelos, especialmente em termos de acurácia direcional e redução de erros.

No WEKA, o desempenho dos modelos foi avaliado por meio de duas abordagens distintas: as abas Classify e Forecast. Os resultados variaram substancialmente entre esses dois modos de avaliação. Notadamente, o algoritmo M5Rules destacou-se como um dos melhores modelos na aba Forecast, atingindo alta acurácia na previsão de valores futuros e demonstrando capacidade de capturar os movimentos direcionais dos preços do BTC. Paradoxalmente, quando avaliado na aba Classify, mesmo com a partição dos dados respeitando a ordem temporal, o M5Rules apresentou coeficiente de correlação próximo de zero, revelando sua limitação em modelar de forma eficaz a totalidade da série histórica nesse formato.

Essa divergência pode ser explicada pela própria natureza do algoritmo M5Rules e pela volatilidade intrínseca e mudanças estruturais presentes na série temporal do BTC. Embora a



linearidade temporal tenha sido preservada no processo de treinamento e teste, a abordagem baseada em regras segmentadas possivelmente enfrentou dificuldades para generalizar diante das frequentes mudanças de regime e dos comportamentos estocásticos da série. Em contraste, ao operar de forma sequencial (como ocorre na aba Forecast), o modelo pareceu adaptar-se melhor a padrões locais, sem a necessidade de representar transições abruptas do passado.

Comparativamente, o modelo de Regressão Linear também demonstrou ganhos substanciais quando implementado no WEKA, apresentando valores consistentemente menores nas três medidas avaliadas, sendo respectivamente 2.92% de erro percentual, RMSE de 1583.9804 e MAE igual à 1161.2834. Isso sugere que modelos mais simples podem obter bons desempenhos quando adequadamente ajustados aos efeitos de tendência e implementados em ambientes estatísticos mais robustos.

De forma similar, o M5P, estruturalmente semelhante ao M5Rules, mantiveram-se com erros menores na plataforma WEKA em relação ao R. O modelo SMOReg apresentou um valores ótimos em todas as medidas no WEKA, sendo o segundo com o melhor desempenho, a destacar seu baixo MAPE de 2,40% e RMSE de 1450.2472, sendo bem próximos do valores obtido pelo M5Rules.

Modelos baseados em redes neurais, como o Multilayer Perceptron (MLP), apresentaram desempenho substancialmente superior no R em todas as medidas avaliadas, o que destaca a vantagem das bibliotecas do R na manipulação de relações complexas e não-lineares características de séries financeiras. Em contraste, o WEKA apresentou instabilidades significativas ao executar MLP, com travamentos frequentes e problemas de alocação de memória.

De forma geral, as diferenças entre plataformas influenciaram diretamente os resultados preditivos, com o R apresentando desempenho consistentemente superior na maioria das medidas de desempenho analisadas. Contudo, os melhores resultados usando as referidas medidas, para este conjunto de dados se obteve no WEKA, com M5Rules e SMOReg, respectivamente, apresentando os melhores desempenhos.

CONSIDERAÇÕES FINAIS

Este estudo avaliou a usabilidade e o desempenho preditivo da plataforma WEKA na previsão da série temporal de preços do BTC, comparando seus resultados com aqueles obtidos no ambiente R. Os achados demonstraram que, embora o WEKA ofereça uma estrutura de modelagem simples e acessível, sua eficácia na previsão de séries financeiras complexas e

voláteis como o BTC é limitada. Apesar de alguns algoritmos como o M5Rules apresentarem resultados competitivos no módulo de previsão, os modelos do WEKA geralmente tiveram desempenho inferior quando avaliados em termos de consistência e confiabilidade na classificação.

Por outro lado, o ambiente R demonstrou maior robustez e precisão, especialmente em métodos baseados em árvores e redes neurais, proporcionando a redução mais consistente dos erros. Os testes estatísticos aplicados ainda caracterizaram a série do BTC como não estacionária, com tendências sistemáticas e heterogeneidade estrutural marcante, o que acentua os desafios enfrentados pelos algoritmos preditivos.

Os resultados indicam que, embora o WEKA possa ser utilizado para exercícios exploratórios ou fins educacionais em previsão, sua aplicabilidade prática em contextos financeiros de alta complexidade é inferior para a maior parte dos algoritmos quando comparado a plataformas estatísticas mais especializadas, como o R.

Dada a bem-sucedida modelagem de um ativo altamente complexo como o Bitcoin, futuras pesquisas poderiam estender essa abordagem a mercados de criptomoedas mais recentes e específicos, como os tokens voltados para jogos blockchain. Esses ativos geralmente apresentam padrões comportamentais distintos, incluindo menor capitalização, maior dinâmica especulativa e movimentos de preços impulsionados por nichos, oferecendo um campo fértil para avaliar ferramentas de previsão sob condições estruturais alternativas.

A presente etapa demonstrou que, apesar de suas limitações, o WEKA apresenta desempenho satisfatório em alguns modelos quando bem calibrados, como o SMOreg e o M5Rules. Esses achados validaram o uso contínuo da plataforma nas próximas fases da pesquisa, especialmente na modelagem preditiva de altcoins. Assim, este artigo se posiciona como elo fundamental entre a etapa exploratória inicial com o BTC e a fase de previsão cruzada entre ativos, consolidada nos artigos seguintes da coletânea.

REFERÊNCIAS

ALI, A. *et al.* Network intrusion detection leveraging machine learning and feature selection. In: IEEE. 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET), 2020, [S.l.]. **Anais [...]**. Piscataway: IEEE, 2020. p. 49–53.

BEHNOOD, A. *et al.* Prediction of the compressive strength of normal and high-performance concretes using M5P model tree algorithm. **Construction and Building Materials**, v. 142, p. 199–207, 2017.

BREIMAN, L. Bagging predictors. **Machine Learning**, v. 24, p. 123–140, 1996.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, p. 5–32, 2001.

CARUANA, R. *et al.* Ensemble selection from libraries of models. In: Proceedings of the Twenty-First International Conference on Machine Learning, 2004, [S.l.]. **Anais [...]**. [S.l.]: [s.n.], 2004. p. 18.

CHOWDHURY, R. *et al.* An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning. **Physica A: Statistical Mechanics and its Applications**, v. 551, p. 124569, 2020.

CONTRERAS, R. C. *et al.* Genetic algorithm for feature selection applied to financial time series monotonicity prediction: Experimental cases in cryptocurrencies and Brazilian assets. **Entropy**, v. 26, n. 3, p. 177, 2024.

DERBENTSEV, V. *et al.* Forecasting cryptocurrency prices time series using machine learning approach. In: SHS WEB OF CONFERENCES, 2019, [S.l.]. **Anais [...]**. [S.l.]: EDP Sciences, 2019. v. 65, p. 02001.

DERBENTSEV, V.; MATVIYCHUK, A.; SOLOVIEV, V. N. Forecasting of cryptocurrency prices using machine learning. In: **Advanced Studies of Financial Technologies and Cryptocurrency Markets**. [S.l.]: Springer, 2020. p. 211–231.

DERBENTSEV, V. *et al.* Forecasting cryptocurrency prices using ensembles-based machine learning approach. In: IEEE. International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), 2020, [S.l.]. **Anais [...]**. Piscataway: IEEE, 2020. p. 707–712.

FANG, Q. *et al.* Prediction of blast-induced ground vibration in open-pit mines using a new technique based on imperialist competitive algorithm and M5Rules. **Natural Resources Research**, v. 29, n. 2, p. 791–806, 2020.

FRANK, E.; MAYO, M.; KRAMER, S. Alternating model trees. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing, 2015, [S.l.]. **Anais [...]**. [S.l.]: [s.n.], 2015. p. 871–878.

GNANAMBAL, S. *et al.* Classification algorithms with attribute selection: an evaluation study using WEKA. **International Journal of Advanced Networking and Applications**, v. 9, n. 6, p. 3640–3644, 2018.

HARYADI, D. *et al.* Implementation of support vector regression for polkadot cryptocurrency price prediction. **JOIV: International Journal on Informatics Visualization**, v. 6, n. 1-2, p. 201–207, 2022.

HOROWITZ, J.; KLEMELÄ, J.; MAMMEN, E. Optimal estimation in additive regression models. **Bernoulli**, v. 12, n. 2, p. 271–298, 2006.

KALEKAR, P. S. Time series forecasting using Holt-Winters exponential smoothing. **Kanwal Rekhi School of Information Technology**, v. 4329008, n. 13, p. 1–13, 2004.

- KARGAR, K.; SAFARI, M. J. S.; KHOSRAVI, K. Weighted instances handler wrapper and rotation forest-based hybrid algorithms for sediment transport modeling. **Journal of Hydrology**, v. 598, p. 126452, 2021.
- KOHAVI, R. Wrappers for performance enhancement and oblivious decision graphs. stanford university, 1996.
- KURNIAWAN, K.; MADELAN, S. Forecasting using time series analysis method in cryptocurrency period 2015–2022. **International Journal of Innovative Science and Research Technology**, v. 7, p. 1454–1459, 2022.
- NEWSON, R. Parameters behind “nonparametric” statistics: Kendall’s tau, Somers’ D and median differences. **The Stata Journal**, v. 2, n. 1, p. 45–64, 2002.
- NIRANJAN, A. *et al.* ERCR TV: Ensemble of random committee and random tree for efficient anomaly classification using voting. In: Proceedings of the 2018 3rd International Conference for Convergence in Technology (I2CT), 2018, [S.l.]. **Anais [...]**. Piscataway: IEEE, 2018. p. 1–5.
- PAUDEL, N.; BHATTA, J. Mushroom classification using random forest and REP tree classifiers. **Nepal Journal of Mathematical Sciences**, v. 3, n. 1, p. 111–116, 2022.
- POYRAZ, İ.; GÜRHANLI, A. **Demand forecasting with time series analysis using drug sales data**. International Journey of Engineering Research and Applications, v. 10, n. 7, p. 51–54, 2020.
- QUINLAN, J. R. *et al.* Learning with continuous classes. In: 5th Australian joint conference on artificial intelligence. 1992. p. 343–348.
- RAMCHOUN, H. *et al.* Multilayer perceptron: Architecture optimization and training. **International Journal of Interactive Multimedia and Artificial Intelligence**, v. 4, n. 1, p. 26–30, 2016.
- REDDY, B. H. M.; REDDY, S. V.; SAROJAMMA, B. Data Mining Techniques for estimation of wind speed using WEKA. **International Journal of Computer Sciences and Engineering (IJCSE)**, v. 9, n. 9, p. 49–53, 2021.
- SABANCI, K.; ÜNLERŞEN, M. F.; POLAT, K. Classification of different forest types with machine learning algorithms. **Research in Rural Development**, v. 1, p. 254–260, 2016.
- SAMSON, T. K. Comparative analysis of machine learning algorithms for daily cryptocurrency price prediction. **Information Dynamics and Applications**, v. 3, n. 1, p. 64–76, 2024.
- SCHULZ, E.; SPEEKENBRINK, M.; KRAUSE, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. **Journal of Mathematical Psychology**, v. 85, p. 1–16, 2018.

DOCHARKHEHSAZ, M. *et al.* Investigation of the Differential Power of Young's Internet Addiction Questionnaire Using the Decision Stump Tree. *Computational Intelligence and Neuroscience*, v. 2022, n. 1, p. 3930273, 2022.

SINGH, P.; AGRAWAL, S. Node localization in wireless sensor networks using the M5P tree and SMOreg algorithms. In: 5th International Conference on Computational Intelligence and Communication Networks, 2013, [S.l.]. **Anais [...]**. Piscataway: IEEE, 2013. p. 104–104.

SKOWRON, A.; SURAJ, Z. A parallel algorithm for real-time decision making: a rough set approach. **Journal of Intelligent Information Systems**, v. 7, p. 5–28, 1996.

SU, X.; YAN, X.; TSAI, C.-L. Linear regression. **Wiley Interdisciplinary Reviews: Computational Statistics**, v. 4, n. 3, p. 275–294, 2012.

SUN, G. Cryptocurrency price prediction based on Xgboost, LightGBM and BNN. **Applied and Computational Engineering**, v. 49, p. 273–279, 2024.

WITTEN, I. H.; FRANK, E.; HALL, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3. ed. Burlington: Morgan Kaufmann, 2011.

ZHANG, A.; ZHANG, L. RBF neural networks for the prediction of building interference effects. *Computers & Structures*, v. 82, n. 27, p. 2333-2339, 2004.