

Comparação de Modelos de Previsão de Séries Temporais aplicados às Cotações de Criptomoedas

Felippe Feliciano de Carvalho, Guilherme Costa de Oliveira, Samuel da Cruz Souza, Túlio Wilian Caproni & Marcelo Vinícius Cysneiros Aragão

Abstract— Despite its growing expansion, the cryptocurrency market still faces great distrust from investors due to the high volatility of their prices. This work consists of a comparative study of time series prediction models applied to cryptocurrency quotations. In this sense, a review of the bibliography related to the topic was carried out; then, the functioning of each of the models considered is briefly displayed: Linear Regression, Polynomial Regression, ARIMA, Prophet, and LSTM. Afterward, several experiments are conducted and their respective results collected under the light of metrics known in the literature. Finally, based on the evaluation and comparison of the performance of each model, the best one is indicated for the prediction of cryptocurrency values. Additionally, proposals for improvements and future work are raised.

Index Terms— Machine Learning, Cryptocurrency, Artificial Intelligence, Financial Time Series Forecasting

Resumo— Apesar de sua crescente expansão, o mercado das criptomoedas ainda enfrenta grande desconfiança por parte de investidores devido a alta volatilidade de seus preços. Este trabalho consiste no estudo comparativo de modelos de previsão de séries temporais aplicadas às cotações de criptomoedas. Neste sentido, é feita uma revisão da bibliografia relacionada ao tema; em seguida, são dispostos brevemente o funcionamento de cada um dos modelos considerados: regressão linear, regressão polinomial, ARIMA, Prophet e LSTM. Na sequência, diversos experimentos são conduzidos e seus respectivos resultados coletados sob a luz de métricas conhecidas na literatura. Por fim, a partir da avaliação e comparação do desempenho de cada modelo, o melhor deles é apontado para a previsão de valores das criptomoedas. Adicionalmente, são levantadas propostas de melhorias e trabalhos futuros.

Palavras Chave— Aprendizado de Máquina, Criptomoeda, Inteligência Artificial, Previsão de Série Temporal Financeira

I. INTRODUÇÃO

A criptomoeda é um tipo de moeda digital obtida [por meio] de tecnologias de *blockchain*, que consiste em redes e meios de troca usando criptografia para proteger transações [1] e funções criptográficas que apresentam transparência, descentralização e imutabilidade. Além do Bitcoin (BTC), inventado em 2009, surgiram moedas alternativas como Ethereum (ETH) e Ripple (XRP). [2]

De acordo com a empresa de informações financeiras Economatica, o Bitcoin, uma das principais criptomoedas, foi um dos

investimentos com maior valorização mundial no ano de 2020, partindo de US\$ 7.300 para US\$ 29.433, uma alta de 303%. Em reais, graças a valorização do dólar sobre a moeda brasileira, tal criptomoeda passou de R\$ 29.399 para R\$ 152.700, uma alta de 419%. Atualmente a moeda está sendo cotada em US\$63.252,63, [conforme ilustrado pela Figura 1]. [3][4]

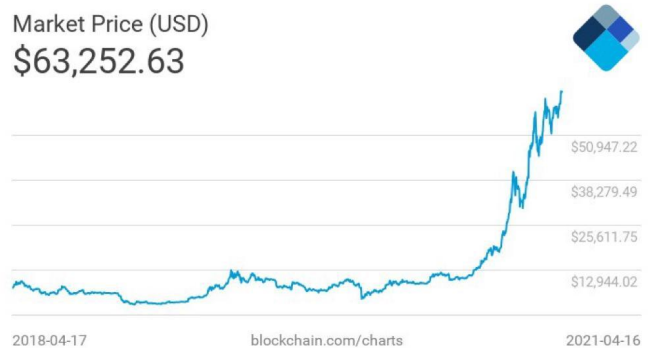


Fig. 1. Preço médio de mercado em dólares americanos nas principais bolsas de Bitcoin [4].

Nesse contexto, é interessante destacar o crescimento expressivo do interesse acadêmico em Bitcoin e *blockchain*. O número de artigos do Google Scholar¹ mencionando Bitcoin atingiu um total de 97.800, sendo 15.800 publicados em 2020; quanto a *blockchain*, foram 205.000 em dezembro de 2020, dos quais 33.800 foram publicações feitas ao longo do ano. [5]

Apesar do crescimento exponencial do mercado de criptomoedas nos últimos anos, ele ainda enfrenta grandes desconfianças dos investidores e usuários graças à alta volatilidade de seus valores, sem períodos de negociação fechados, com capitalização relativamente menor e alta disponibilidade de dados no mercado. [6]

Ainda que haja desconfiança por parte de investidores, o mercado de criptomoedas atrai aqueles com maior experiência, curiosos e entusiastas, devido às potenciais oportunidades de lucro. A variabilidade de preço continua sendo um ponto negativo, principalmente para investidores comuns, dadas as chances de súbita desvalorização, o que o torna um ativo financeiro de alto risco. [7]

Recentemente a Coinbase² lançou suas ações na bolsa americana Nasdaq por meio de uma listagem direta, tornando-se a

Trabalho de Conclusão de Curso apresentado ao Instituto Nacional de Telecomunicações, como parte dos requisitos para a obtenção do Grau de Bacharel em Engenharia da Computação. Orientador: Prof. Me. Marcelo Vinícius Cysneiros Aragão. Trabalho aprovado em [08/07/2021].

¹Plataforma do Google voltada para busca de conteúdo científico. Disponível em <https://scholar.google.com.br/>.

²Maior plataforma para compra e venda de criptomoedas dos Estados Unidos.

primeira empresa do tipo a ter papéis listados. De acordo com Fábio Alves Moura, sócio fundador da AMX Law³, essa abertura de capital dá sinais importantes ao mercado. As empresas de *blockchain* e criptomoedas estão sendo vistas com um nível de seriedade e maturidade diferente, o que vem aumentando a confiança de investidores que nunca tiveram contato com este setor e que podem optar por diversificar suas carteiras, passando também a investir nesta classe de ativos. [8]

Uma abordagem para o problema da alta volatilidade das criptomoedas consiste na aplicação de inteligência artificial baseada em modelos de aprendizado de máquina, com o objetivo de prever os valores de sua cotação. [7]

Dada a dificuldade de decidir o momento ideal e qual tipo de operação realizar (compra ou venda), a previsão de dados de séries temporais financeiras sempre atraiu a atenção do setor financeiro e investidores [7]. Fatores como a aplicação generalizada de sistemas de negociação eletrônicos automatizados, juntamente com a crescente demanda por rendimentos mais elevados, estimula pesquisadores e profissionais na busca pela implementação de modelos melhores. Consequentemente, novas publicações e implementações continuam sendo adicionadas à literatura de finanças e inteligência computacional. [9]

Assim, visto que os movimentos dos preços de mercado não são aleatórios, mas se comportam de forma altamente não linear e dinâmica, a previsão da direção do mercado utilizando modelos de séries temporais pode produzir investimentos mais seguros e/ou lucros maiores. [6]

O presente trabalho foca na previsão de séries temporais aplicadas a cotações de criptomoedas por meio de estudo comparativo entre diversos modelos, como regressão linear, regressão polinomial, ARIMA (do inglês, *Autoregressive Integrated Moving Average*), LSTM (do inglês, *Long Short Term Memory*) e Prophet, que serão abordados nas seções futuras. Tal ação foi realizada a partir da coleta de valores históricos da flutuação cambial da moeda estudada (Bitcoin).

O artigo está dividido em cinco partes: a seção II. aborda uma revisão teórica acerca dos principais temas relacionados ao trabalho; em seguida, a seção III. traz uma revisão bibliográfica, apresentando os principais pontos estudados em outros trabalhos sobre o presente tema; posteriormente, na seção IV. estão dispostas as etapas envolvidas nos experimentos; na seção V. os resultados são apresentados e discutidos, e os modelos são comparados sob a luz de métricas bem disseminadas na literatura, no sentido de apontar qual(is) seria(m) mais indicado(s) para o objetivo do trabalho; por fim, na seção VI. são feitas as considerações finais sobre o trabalho e sugeridas propostas de novas pesquisas.

II. REVISÃO DA TEORIA

A temática envolvida neste trabalho engloba determinadas áreas de estudo, tais como criptomoeda, aprendizado de máquina e previsão de série temporal financeira. A seguir, é feita uma breve revisão teórica de cada uma delas.

Disponível em <https://coinbase.com>.

³Empresa suíça de consultoria jurídica especializada em *blockchain*, direito internacional e direito empresarial avançado. Disponível em <https://pt.amxlaw.com/>

A. Criptomoeda

As criptomoedas encontram-se em um meio totalmente digital. Através de uma rede globalizada, as trocas e transações financeiras são baseadas em funções criptográficas. A criptomoeda pioneira foi o Bitcoin, que surgiu em 2009, tornando-se a moeda mais antiga e conhecida. A partir do Bitcoin, inúmeras outras criptomoedas surgiram no mercado, como a Litecoin, Ripple, Ethereum, entre outras. Atualmente o Bitcoin apresenta a primeira colocação no mercado em relação ao valor, atingindo cotações na ordem de 1 BTC = R\$333 mil.[4]

Uma das principais características das criptomoedas é a alta variação do seu valor, mesmo em períodos curtos de tempo (ao longo de um dia, por exemplo), se comparadas às moedas fiduciárias como o dólar ou o real. Tal particularidade gera bastante desconfiança por parte dos investidores, mas também atrai aqueles que veem esta volatilidade como uma oportunidade de lucro.

A criptomoeda se beneficia da tecnologia *blockchain*, que pode ser entendida como um tipo particular de banco de dados que armazena informação em grupos, denominados blocos, que são preenchidos e interligados ao bloco completo anterior. Tal tecnologia garante transparência, imutabilidade e uma das características mais relevantes, a descentralização. Isso a torna virtualmente imune a interferências (governamentais, por exemplo), já que a descentralização garante justamente que ela não seja controlada por uma autoridade central.

A transação de criptomoedas se dá por meio do uso de chaves privadas ou públicas. Essas ações podem ser feitas com taxas de processamento mínimas, permitindo que os usuários evitem taxas muitas vezes abusivas cobradas por instituições financeiras tradicionais.

B. Aprendizado de Máquina

Aprendizado de máquina é um campo de estudos na área de Inteligência Artificial (IA) que busca compreender, generalizar e interagir com o mundo através de dados. A característica de inferência pode, por exemplo, tentar prever eventos e sugerir ações com base no que aconteceu no passado e no que está acontecendo no presente. Existem vários tipos de modelos para essa área como os exemplos a seguir:

- Árvores de decisão: são modelos de aprendizado de máquina supervisionado não-paramétricos, frequentemente utilizados em tarefas de classificação e regressão. [10]
- Regressão bayesiana: consiste em um método de inferência estatística aplicado para complementar estatísticas frequentistas, pois formula a regressão linear a partir de distribuições de probabilidades em vez de estimativas pontuais. [11]
- Máquina de vetores de suporte: método de aprendizado de máquina supervisionado capaz de resolver problemas de classificação, regressão e agrupamento inclusive em casos não-lineares. [12]
- Redes neurais artificiais: constituem-se em técnicas computacionais que apresentam um modelo matemático baseado na estrutura neural de organismos inteligentes, de modo a obter conhecimento através da experiência. Uma

grande rede neural artificial pode ter centenas ou milhares de unidades de processamento. [13]

C. Previsão de Série Temporal Financeira

A variação do preço das criptomoedas ao longo do tempo pode ser vista como uma série temporal financeira. A previsão deste tipo de série, ilustrada na Figura 2 é um problema clássico, e métodos tradicionais vem sendo aplicados à previsão dos preços e variações de moedas convencionais (fiduciárias), valor de ações ou preço de algum produto [2]. Um dos modelos mais utilizados para esta finalidade é o modelo ARIMA, mas outros como LSTM, regressão linear, regressão polinomial e Prophet também aplicam-se.

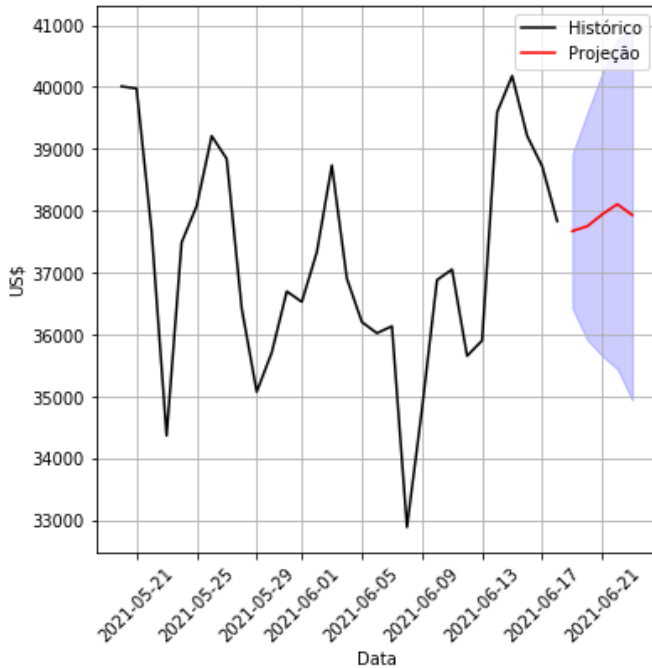


Fig. 2. Exemplo de previsão de série temporal na qual os eixos horizontal e vertical representam o tempo e os valores associados, respectivamente.

c.1. Regressão Linear

A regressão linear é um método estatístico que utiliza a relação entre duas ou mais variáveis, de modo que uma variável possa ser estimada a partir da(s) outra(s) através de um polinômio de primeiro grau [14], como ilustra a Figura 3.

c.2. Regressão Polinomial

A regressão Polinomial realiza a previsão de dados de forma semelhante com a regressão linear. Entretanto, o método consiste em estabelecer um polinômio de n -ésimo grau que descreve o comportamento de uma série temporal, conforme exemplificado na Figura 4.

c.3. ARIMA

O modelo ARIMA (do inglês, *Autoregressive Integrated Moving Average*) é uma generalização do modelo ARMA (do

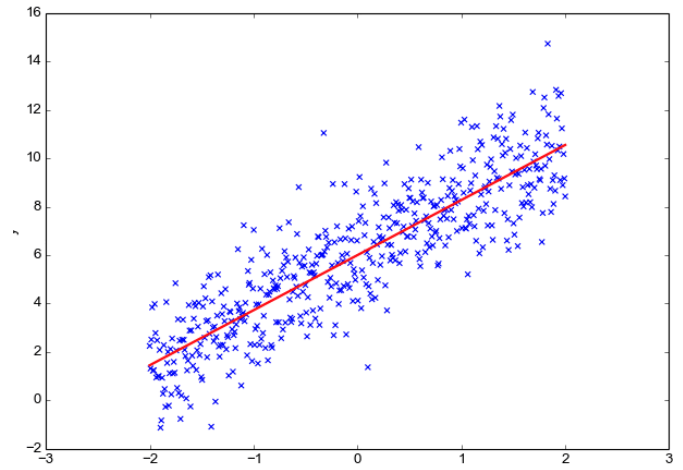


Fig. 3. Exemplo de regressão linear. A reta em vermelho tenta aproximar a distribuição dos dados em azul. [15]

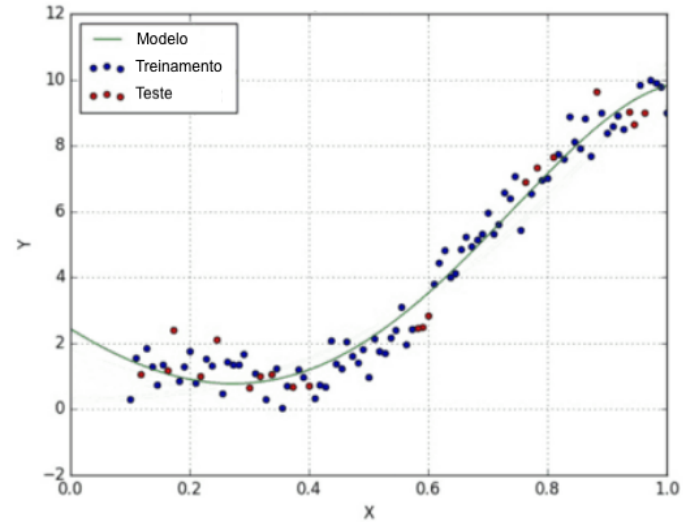


Fig. 4. Exemplo de regressão polinomial. A curva em verde tenta aproximar a distribuição dos dados em azul. [15] (adaptado)

inglês, *Autoregressive Moving Average*), que pode ser particularmente eficaz nos casos em que os dados mostram evidências de não estacionariedade⁴ [16].

O modelo ARIMA é dividido em três partes: AR (do inglês, *Auto Regressive*), parte regressiva em que a variável que evolui e retorna a seus próprios valores anteriores; MA (do inglês, *Moving Average*), a parte da média móvel que indica o erro de regressão, a partir de uma combinação linear com os valores dos erros passados e por último, o I (do inglês, *Integrated*), parte integrada que mostra os valores substituídos com a diferença entre seus valores e os anteriores. Esse processo pode ser repetido mais de uma vez e cada parte do modelo tem o objetivo de ajustar os dados com a melhor combinação possível. Alguns modelos do ARIMA são:

- ARIMA não estacionários: são denominados como ARIMA (p, d, q) , onde p , d e q são números não negativos, sendo p o número de defasagens do modelo, d o número de vezes em que os dados tiveram valores anteriores sub-

⁴Uma série temporal é dita estacionária quando suas características estatísticas são constantes ao longo do tempo.

traídos e q a ordem do modelo de média móvel. [17]

- ARIMA sazonal: são denominados como ARIMA $(p, d, q)(P, D, Q)m$, onde m se refere ao número de períodos de cada estação. Já as letras maiúsculas P, D e Q correspondem à autorregressão, diferença e o movimento médio para parte sazonal do modelo ARIMA.
 - SARIMAX (do inglês, *Seasonal Autoregressive Integrated Moving Average With Exogenous Factors*): é utilizado em conjuntos de dados compostos por ciclos sazonais. As diferenças entre ARIMA e SARIMAX são a sazonalidade e os fatores exógenos, ou seja, o SARIMAX é um modelo que busca prever os valores de uma série temporal cujos valores estão vulneráveis à mudanças vindas de fora do modelo e o próprio modelo não é capaz de explicar tais mudanças. [18]

c.4. LSTM

As redes LSTM (Memória Longa de Curto Prazo), inicialmente propostas por Hochreiter e Schmidhuber [19], são um tipo de rede neural recorrente que utiliza *Deep Learning* (DL) e são compostas por unidades capazes de aprender dependências de longo prazo [20]. Sua topologia é ilustrada na Figura 5.

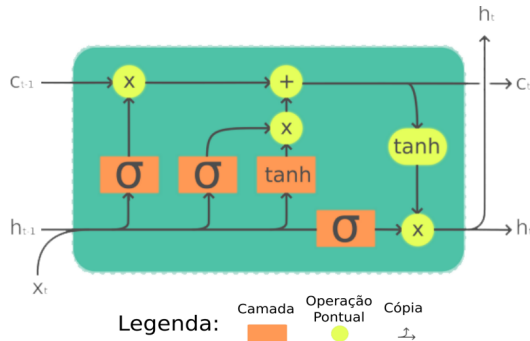


Fig. 5. Esquema de uma rede neural LSTM. [21] (adaptado)

Uma definição de uma RNN (do inglês *Recurrent Neural Network*) [22] é descrita pelas equações 1 e 2, nas quais t representa a época atual, h o nome do estado oculto, W são pesos, X e y são entrada e saída (respectivamente) e σ é a função sigmoide.

$$h^{(t)} = \sigma(W_h x + W_r y^{(t-1)}) \quad (1)$$

$$y = \sigma(W_y h^t) \quad (2)$$

Estes valores se mantêm para as equações (3), (4) e (5) listadas a seguir sobre os estados da rede LSTM.

Segundo Vassali [22], as unidades LSTM são compostas por um neurônio responsável por armazenar o estado da rede. Este, por sua vez, pode ser ativado (ou não) de acordo com a soma ponderada dos portões (*gates*). Os estados podem ser:

- Oculto: determina o que poderá ser esquecido e o que deve entrar ou sair no próximo passo;
- Entrada: combinação do valor atual com o estado oculto;
- Interno: valores com a função de memória.

Os portões, por sua vez, podem ser dos tipos:

- Entrada: toma a decisão para que o portão alcance o estado interno conforme a equação (3);

$$i^{(t)} = \sigma(W^{ix} x^{(t)} + W^{ih} h^{(t-1)} + b_i) \quad (3)$$

- Esquecimento: decide se o estado interno irá desconsiderar o estado anterior representado pela equação (4);

$$f^{(t)} = \sigma(W^{fx} x^{(t)} + W^{fh} h^{(t-1)} + b_f) \quad (4)$$

- Saída: tem o papel de determinar se o estado interno é enviado à saída e ao estado oculto no próximo passo, dado pela equação (5);

$$o^{(t)} = \sigma(W^{ox} x^{(t)} + W^{oh} h^{(t-1)} + b_o) \quad (5)$$

A biblioteca escolhida para o uso do LSTM é a `tensorflow.python.keras`⁵, escrita na linguagem de programação Python. Os parâmetros a serem considerados estão listados a seguir, seguindo a documentação oficial [23]:

- `input shape`: dimensão a qual serão utilizados x dia(s) para a previsão de y dia(s). Exemplo: `120 x 1`;
- `units`: número inteiro positivo, que representa a dimensão do espaço de saída. Exemplo: `50`;
- `activation`: nome da função de ativação. Exemplo: `relu`;
- `return sequences`: *booleano* para retornar (ou não) a última saída. Exemplo: `True`;

c.5. Prophet

O Prophet é uma ferramenta computacional de código aberto mantida pelo Facebook e pode ser utilizado para prever dados de séries temporais com base em modelos aditivos que possuem tendência não linear e podem ser ajustados com sazonalidade anual, semanal e diária [24]. Segundo R. Castelão, o Prophet é otimizado para realizar previsões na área de negócios, tomando como base tarefas encontradas no Facebook [25].

Através de um modelo de regressão aditivo com uma curva linear ou logística, o Prophet realiza suas projeções incluindo um componente sazonal anual que é modelado utilizando uma série de Fourier e um componente sazonal semanal utilizando variáveis *dummy*.

Os parâmetros a serem considerados estão descritos na documentação [26], os mais relevantes para a pesquisa estão listados a seguir:

- `interval width`: porcentagem de dados utilizados para treino. Exemplo: `0.80`;
- `holidays`: Datas a serem consideradas como feriados. Exemplo (para não considerar feriados): `None`;
- `growth`: especificação do tipo “linear” ou “logístico”. Exemplo: `linear`;
- `changepoints`: datas nas quais podem incluir pontos de mudança em potencial. Exemplo (caso não existam tais datas): `None`;
- `n changepoints`: número inteiro de potenciais pontos de mudanças a serem incluídos. Exemplo: `25`;

⁵Documentação oficial da biblioteca Tensorflow. Disponível em https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM.

- `daily seasonality`: ajuste da sazonalidade diária. Exemplo: `False`;
- `weekly seasonality`: ajuste da sazonalidade semanal. Exemplo: `auto`;
- `yearly seasonality`: ajuste da sazonalidade anual. Exemplo: `auto`;
- `seasonality mode`: tipo da sazonalidade (aditiva ou multiplicativa). Exemplo: `multiplicative`.

III. REVISÃO DA BIBLIOGRAFIA

O objetivo deste trabalho é avaliar e comparar o desempenho de diferentes modelos de previsão de séries temporais aplicados a cotações de criptomoedas. Assim, nesta seção são apresentados e discutidos artigos, pesquisas e estudos encontrados na literatura atual acerca do tema.

A. *Investigating the Problem of Cryptocurrency Price Prediction: A Deep Learning Approach* (2020)

O estudo de Pintelas et al. [2] avaliou o desempenho de algoritmos avançados de DL para prever o preço e o movimento das três criptomoedas mais populares: bitcoin, [ethereum] e ripple. Os objetivos foram investigar se o DL pode prever os preços de criptomoedas de forma eficaz; se os preços seguem um caminho aleatório e se existe um método de avaliação adequado de modelo de previsão de criptomoedas. Para isso foram avaliados modelos de DL: CNN (do inglês, *Convolutional Neural Networks*), LSTM e BiLSTM (do inglês, *Bidirectional LSTM*). Estes também foram comparados com modelos de aprendizado de máquina tradicionais. De acordo com os autores, dado o nível de complexidade do problema, os modelos apresentados são preditores de preço de criptomoeda ineficientes e não confiáveis. Modelos como LSTM e CNNs com técnicas avançadas de DL não foram capazes de resolver de forma eficiente. A partir do estudo, também concluíram que os preços das criptomoedas seguem um processo aleatório, com provavelmente poucos padrões ocultos, de modo a dificultar sua identificação por estruturas inteligentes e impedir a formação de previsões precisas e confiáveis.

B. *Forecasting of Cryptocurrency Prices Using Machine Learning* (2020)

Derbentsev, Matviychuk e Soloviev [27] consideram a aplicação de métodos tradicionais, como modelos causais ou modelos clássicos de séries temporais, um meio ineficaz para a previsão de criptomoedas. Desse modo, realizaram um trabalho com objetivo de comparar as propriedades de previsão de curto prazo das taxas de câmbio de criptomoedas dentre diferentes métodos de aprendizado de máquina: algoritmo BART (do inglês, *Bayesian Additive Regression Trees*), redes perceptron multicamadas (MLP – do inglês, *Multilayer Perceptron*), um conjunto de árvores de classificação e regressão (do inglês, CART – *Classification and Regression Trees* e florestas aleatórias (RF – do inglês, *Random Forests*). Através deles, em um espaço de tempo de 90 dias, a dinâmica das três criptomoedas mais capitalizadas (Bitcoin, Ethereum e Ripple) foi estimada.

A partir da análise comparativa da capacidade preditiva dos modelos construídos, verificou-se que todos os modelos descrevem adequadamente a dinâmica das criptomoedas com o erro percentual absoluto médio – para os modelos BART e MLP em média 3,5%, e para modelos RF dentro de 5%. Para prever a direção de uma mudança no preço ou tendência, em vez de seu valor numérico, a aplicação prática do modelo BART também foi demonstrada na previsão da direção da mudança no preço por um período de 90 dias. Para tanto, um modelo de classificação binária foi utilizado na metodologia de avaliação do grau de atratividade das criptomoedas. Simulações confirmaram a viabilidade do uso de métodos e modelos de aprendizado de máquina para a previsão de curto prazo de séries temporais financeiras. Os autores concluíram que os modelos e seus conjuntos podem ser a base para os algoritmos de sistemas de negociação automatizados para negociação pela Internet.

C. *Cryptocurrency forecasting with deep learning chaotic neural networks* (2019)

Lahmire e Bekiros [28] realizaram um estudo experimental, cujo objetivo foi avaliar a previsibilidade das moedas digitais ativas Bitcoin, Digital Cash e Ripple e examinar sua dinâmica não linear inerente, incluindo caoticidade e fractalidade. Além disso, visaram construir uma rede neural complexa com base em DL para extrair automaticamente padrões ocultos, revelando a dinâmica não linear de suas séries temporais de modo a realizar uma análise pragmática. Segundo os autores, resultados decorrentes de uma perspectiva de dinâmica não linear devem indicar se as moedas digitais estudadas são previsíveis ou não a curto prazo, dependendo da fractalidade medida e caoticidade, enquanto resultados da introdução do DL demonstraram a consistência e a precisão de sua capacidade de previsão.

O maior expoente de Lyapunov (MLE – do inglês, *Maximal Lyapunov Exponent*) foi utilizado para verificar a existência de mapeamentos determinísticos não-lineares e uma análise de flutuação sem tendência (DFA – do inglês, *Detrended Fluctuation Analysis*) com base no expoente de Hurst extraído da série temporal para detectar caos e/ou características fractais das moedas digitais subjacentes. O modelo de DL escolhido foi LSTM. Este também foi comparado com uma rede neural de regressão generalizada (GRNN – do inglês, *General Regression Neural Network*) para testar consistência e robustez. Como resultado, a partir dos testes realizados [verificou-se] que as séries temporais de todas as moedas digitais exibem dinâmica fractal, memória longa e autossimilaridade. A previsibilidade do LSTM foi significativamente maior quando comparada à GRNN, pois o último falhou em aproximar os padrões ocultos não lineares globais, independentemente do grau de contaminação com ruído, uma vez que são baseados em *kernels* gaussianos adequados apenas para aproximação local de sinais não estacionários. Por fim, os autores consideraram a aplicação do DL altamente eficiente para este fim de pesquisa.

D. *Financial time series forecasting with deep learning : A systematic literature review: 2005-2019* (2020)

O estudo de Sezer, Gudelek e Ozbayoglu [9] consiste em uma revisão sistemática da literatura com objetivo de fornecer o estado atual da pesquisa de implementações de DL para a previsão

de séries temporais financeiras. Os autores buscaram identificar quais modelos são usados para a previsão de séries temporais financeiras, [compará-los] com os modelos de aprendizado de máquina tradicionais e indicar possíveis direções futuras de tal técnica. Para isso, com foco em implementações de DL para previsão de séries temporais financeiras, revisaram jornais e conferências, teses de mestrado e doutorado, capítulos de livros e artigos publicados em inglês entre os anos de 2005 a 2019.

Como resultado, os autores estabeleceram que modelos baseados em RNN (Rede Neural Recorrente), com destaque ao LSTM, são os mais utilizados para a previsão de séries temporais. Já em relação ao desempenho, na maioria dos estudos, os modelos de DL foram superiores aos de aprendizado de máquina. As razões para a crescente preferência por implementações de DL podem ser explicadas pelos avanços na capacidade de computação, disponibilidade de Big Data, desempenho superior na previsão de séries temporais financeiras, capacidade de detectar recursos implícitos e o fato de ser um modelo mais recente. Futuramente, relatam que NLP (Processamento de linguagem natural), semântica e modelos híbridos baseados em mineração de texto combinados com dados de série temporal poderão ser mais comumente usados. A partir disso, concluíram que apesar de as previsões financeiras serem uma área com longa história de pesquisa, o surgimento de novos modelos de DL tem aumentado o interesse e oportunidades aos pesquisadores.

E. *Predição de Séries Temporais Aplicada ao Mercado de Criptomoedas (2019)*

Em vista da alta volatilidade das criptomoedas, o estudo de Almeida [7] empregou modelos de aprendizado de máquina a fim de prever os valores de cotação do Bitcoin. Para isso, a partir da série histórica do Bitcoin, foi desenvolvido um modelo utilizando rede neural recorrente (RNN – do inglês, *Recurrent Neural Network*) para realizar uma previsão do melhor momento de comercialização (compra e venda) de Bitcoin, utilizando a combinação da curva bruta da criptomoeda com indicadores técnicos: SAR (do inglês, *Stop and Reversal*), Médias Móveis Simples (SMA – do inglês, *Simple Moving Average*) e Índice de Força Relativa (RSI – do inglês, *Relative Strength Index*).

O desenvolvimento do algoritmo consistiu na extração, modelagem e análise de dados. A partir da plataforma [de código aberto] CryptoDataDownload, foi extraída a base de dados Bitsamp, cujos dados utilizados são relativos à paridade BTCUSD (indicador que relaciona Bitcoin (BTC) e dólares americanos (USD)) no período de 28/11/2014 até 29/11/2019. Feita a avaliação e combinação de inúmeros indicadores para averiguar a qualidade da geração do modelo, definiu-se que o melhor resultado obtido foi o indicador de curva (valor de fechamento diário) com uma raiz quadrada do erro quadrático médio de 260.137, sendo esse valor um parâmetro para pesquisas futuras. Assim, o autor conclui que as redes neurais recorrentes, podem ser uma ferramenta promissora para a resolução do problema em questão.

F. *Comparative performance of machine learning algorithms for cryptocurrency forecasting (2018)*

Em vista do surgimento de inúmeros métodos para a construção de modelos incluindo aprendizado de máquina, Hitam e Ismail [29] realizaram uma comparação do desempenho de algoritmos para a previsão de criptomoedas com foco na previsão de séries temporais. As moedas analisadas foram Bitcoin, Ethereum, Litecoin, Nem, Ripple e Stellar. Também foram considerados estudos sobre aprendizado de máquina e modelos como máquinas de Vetores de Suporte (SVM – do inglês, *Support Vector Machines*), redes neurais artificiais e DL.

Os classificadores sobre a criptomoeda selecionada verificaram que o SVM apresentou uma precisão de 95,5%, sendo assim superior aos demais. A partir da análise comparativa entre o valor previsto pelo aprendizado de máquina versus o valor real, o SVM tem valores comparáveis para todas as criptomoedas para o período de 17/01/2017 a 16/01/2018. Além disso, o resultado é mais explorado usando o cálculo do MAPE, onde os resultados demonstraram que SVM possui o menor valor de MAPE. Desse modo, o SVM é considerado um modelo de previsão confiável para a criptomoeda selecionada. Os autores ainda pretendem aprimorar o algoritmo na taxa de precisão do preço previsto, realizar um trabalho futuro a fim de otimizar ainda mais o SVM e obter o resultado mais preciso de acordo com o valor real.

G. *Forecasting cryptocurrency prices time series using machine learning approach (2019)*

Em vista da crescente popularidade da aplicação de modelos de aprendizado de máquina para a previsão do preço de criptomoedas, Derbentsev et al. [17] realizaram um estudo com objetivo de construir um modelo de previsão à curto prazo dos preços das três criptomoedas mais capitalizadas: Bitcoin, Ethereum e Ripple. Para isso, os autores optaram pela aplicação da análise de séries temporais utilizando a metodologia CRISP-DM (do inglês, *Cross Industry Standard Process for Data Mining*). Assim, utilizou-se o modelo BART, uma generalização de modelos padrão de árvores de regressão adaptado para dados de séries temporais. A fim de comparar a capacidade preditiva do algoritmo BART, também foi feita uma previsão utilizando os modelos clássicos ARIMA e ARFIMA (do inglês, *Autoregressive Fractionally Integrated Moving Average*). O tamanho da amostra para treinamento para todos os subperíodos do algoritmo BART foi de 80% do tamanho total da amostra e 20% foi usado como conjunto de dados fora da amostra. O aplicativo de nuvem Microsoft Azure Machine foi escolhido para a implementação dos modelos. A previsão foi realizada em cinco horizontes de tempo diferentes: 5, 10, 14, 21 e 30 dias usando três modelos para cada criptomoeda. Já para estimar as propriedades prognósticas dos modelos, utilizou-se a métrica RMSE. Os resultados demonstraram uma maior precisão do algoritmo BART para todas as séries temporais de criptomoedas e subperíodos analisados quando comparado aos modelos ARIMA-ARFIMA tanto nos períodos de lenta ascensão (queda) quanto nos períodos de dinâmica de transição (mudança de tendência).

H. Sentiment-Based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model (2019)

A partir da premissa de que o mercado de criptomoedas atrai predominantemente investidores de varejo, Li et al. [30] realizaram uma análise dos sinais do Twitter como meio de sentimento do usuário para prever as flutuações de preço de uma criptomoeda alternativa, a ZClassic. Para isso foram extraídos *tweets* de hora em hora por 3,5 semanas, classificando-os como positivo, neutro ou negativo. Tais dados foram compilados em um índice de sentimento por hora e junto das somas brutas de sentimento (positivo/negativo/neutro), justapostos a cerca de 400 pontos de dados de preço por hora a fim de treinar um modelo de árvore de regressão de impulso de gradiente extremo. As previsões de preços resultantes deste modelo foram comparadas aos dados históricos de preços, com as previsões resultantes tendo uma correlação de 0,81 com os dados de teste.

Em vista dos resultados, os autores concluíram que o modelo de árvore de regressão de impulso de gradiente externo é capaz de prever as flutuações de preço no mercado de criptomoeda ZClassic e evidenciam o potencial do uso de mídias sociais nesse meio. Contudo, relatam a necessidade de melhora do modelo e testa-lo por um período de tempo mais longo e com outras criptomoedas.

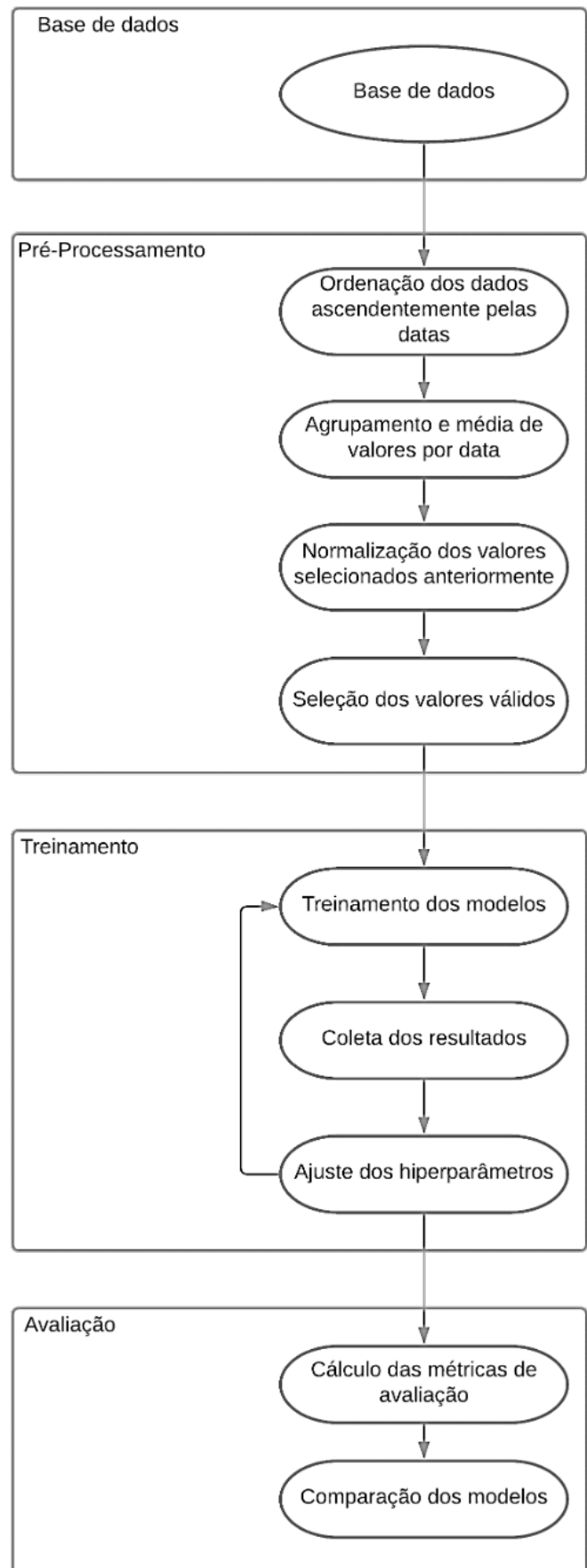
Os trabalhos revisados nesta seção apresentaram diversas abordagens e modelos para a previsão de séries temporais de criptomoedas, além de discorrerem sobre tal mercado. Foram descritas, analisadas e comparadas diferentes técnicas de aprendizado de máquina. Os modelos foram avaliados sob diferentes conjuntos de dados e obtiveram resultados variados.

IV. PROPOSTA

Neste trabalho é proposto um estudo comparativo de diferentes modelos para a previsão de séries temporais aplicadas a criptomoedas. [O Bitcoin, uma das principais criptomoedas, foi escolhido para passar pelas etapas discriminadas no diagrama da Figura 6.] Para a análise comparativa, a etapa de treinamento dos modelos será realizada com cada um dos modelos de detecção considerados.

A. Etapas de Processamento

1. Base de dados: a base de dados é extraída com o auxílio da API (do inglês, *Application Programming Interface*) Crypto Compare⁶, que disponibiliza dados de mercado e série histórica de diversas criptomoedas, dentre elas a Bitcoin.
2. Pré-processamento: consiste na realização dos tratamentos necessários na base de dados que antecedem a entrada no treinamento. Divide-se em:
 - (a) Ordenação dos dados ascendentemente de acordo com as datas.
 - (b) Agrupamento e média de valores por data: os dados são coletados hora a hora ao longo do dia (24 horas), agrupados e substituídos pelo valor de sua média. Esta etapa diminui consideravelmente o volume de



⁶Site que oferece gráficos e cotações ao vivo de diversas criptomoedas. Disponível em <https://min-api.cryptocompare.com>.

Fig. 6. Diagrama de blocos com todas as etapas envolvidas.

dados, permitindo que os modelos sejam treinados mais rapidamente.

- (c) Normalização dos valores selecionados anteriormente: remoção de valores fora do padrão.
 - (d) Seleção dos valores válidos: remoção de valores ausentes, nulos e/ou negativos.
3. Treinamento dos modelos: consiste no ajuste dos parâmetros dos modelos considerados mediante a apresentação dos dados coletados e execução dos algoritmos de treinamento. Os modelos utilizados foram: regressão linear, regressão polinomial, SARIMAX, LSTM e Prophet. Tal processo ocorre de forma independente para cada um dos modelos, a fim de que posteriormente seja possível compará-los através de métricas específicas.
 4. Coleta dos resultados: a partir dos resultados obtidos no treinamento, é feita a coleta desses para posterior análise e ajuste.
 5. Ajuste dos hiperparâmetros: [\[visa alterar aspectos particulares com o intuito de potencializar a capacidade preditiva dos modelos.\]](#) Para isso, as etapas de treinamento, coleta dos resultados e ajuste dos hiperparâmetros são repetidas para que o modelo fique o mais preciso possível (considerando custo computacional e tempos de execução praticáveis).
 6. Cálculo das métricas de avaliação: elas servem como parâmetros de comparação, e torna possível indicar qual(is) modelo(s) apresenta(m) melhor desempenho. As métricas utilizadas foram:
 - MAPE (do inglês, *Mean Absolute Percentage Error*): o erro percentual absoluto médio consiste em um valor de ponto flutuante não negativo no qual o melhor valor possível é 0,0. Não há limite para a magnitude do erro, especialmente quando os valores reais estão próximos de zero [16]. Esta métrica é dada pela Equação (6), na qual A_t é o valor real, F_t é o valor da previsão e n é a quantidade de elementos.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (6)$$

- MAE (do inglês, *Mean Absolute Error*): calcula a diferença entre o valor histórico e o valor projetado [16], onde A_t representa o valor histórico, F_t o valor projetado e n a quantidade de elementos. Quanto mais próximo de 0, mais preciso é o modelo, conforme apresentado na Equação 7.

$$MAE = \frac{1}{n} \sum_{i=1}^n |A_t - F_t| \quad (7)$$

- MSE (do inglês, *Mean Squared Error*) denota a média dos erros quadráticos entre os valores observados e as projeções [16]. Quanto menor o seu valor, melhor a precisão do modelo segundo a Equação 8.

$$MSE = \frac{1}{n} \sum_{i=1}^n (A_t - F_t)^2 \quad (8)$$

- RMSE (do inglês, *Root Mean Squared Error*) representa a raiz quadrada do erro quadrático médio

dos erros entre os valores observados (reais) e as previsões (hipóteses) [16].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (A_t - F_t)^2} \quad (9)$$

- Tempo: corresponde ao intervalo de tempo que o modelo demandou para ser treinado e testado.
7. Comparação dos modelos: nesta etapa, os resultados obtidos a partir de cada um dos modelos escolhidos são comparados. Assim, são usadas as métricas de avaliação citadas anteriormente para avaliar cada modelo. Com a implementação de cada uma das etapas descritas, [\[obteve-se\]](#) diferentes resultados para cada um dos modelos usados. Por meio dos resultados obtidos, é possível realizar uma comparação para analisar qual(is) a(s) melhor(es) alternativa(s) para a previsão das cotações de criptomoedas.

V. EXPERIMENTOS

A. Metodologia

Tomando como referência o [\[estudo de\]](#) Pietro [21], todo o código foi construído em linguagem de programação Python. A API Crypto Compare foi utilizada para a obtenção dos dados históricos da criptomoeda Bitcoin (BTC) e as principais bibliotecas usadas na previsão [\[foram\]](#):

- [\[sklearn\[31\]\]](#): para detecção de *outliers* e ajustes de modelos;
- [\[pmdarima\[32\]\]](#): para os modelo autorregressivo (SARIMAX);
- [\[tensorflow\[23\]\]](#): para modelos de DL (LSTM);
- [\[fbprophet\[24\]\]](#): utilizado para o modelo Prophet;
- [\[tslearn\[33\]\]](#): utilizado em agrupamentos;
- [\[scipy\[34\]\]](#): utilizado para ajustes paramétricos.

a.1. API

A API escolhida provê os dados históricos da moeda virtual Bitcoin com valores de hora em hora, desde o seu início (em meados de 2010) e seu respectivo valor em dólares americanos (US\$). Ao todo, foram coletadas mais de 80 mil observações da série temporal. Entretanto, a fim de se obter uma maior precisão nos modelos, tais dados passaram por um pré-processamento antes de serem utilizados.

a.2. Base de Dados

Os dados entre 2012 e 2017 (representados na Figura 7) foram descartados, pois comparados aos dados atuais, representam um valor constante que poderia interferir na capacidade preditiva dos modelos.

Os dados selecionados são apresentados na Figura 8, em mesma escala que o gráfico anterior. Assim, cerca de 40 mil observações foram descartadas.

Após a seleção do intervalo de tempo com valores promissores, foi feita a remoção de *outliers* conforme ilustrado na Figura 9. De modo geral, 5% dos dados válidos são classificados como [\[outliers\]](#), representando assim cerca de duas

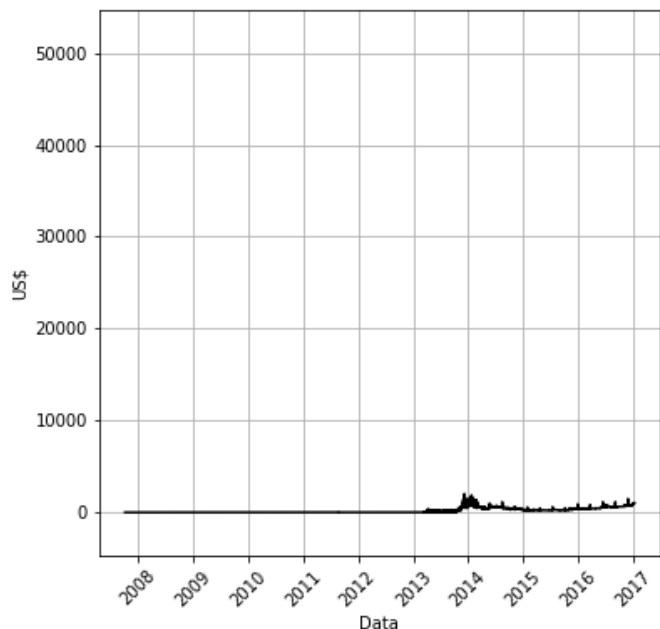


Fig. 7. Dados do período descartado.

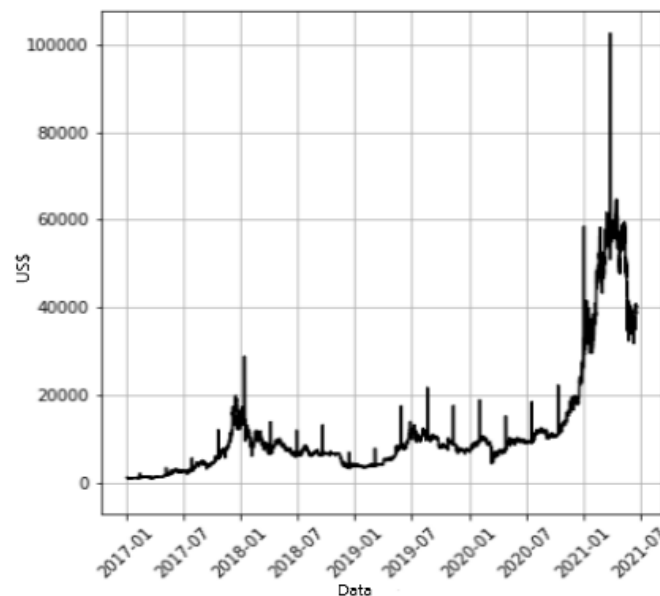


Fig. 8. Dados do período considerado (treinamento e teste).

mil observações. Tais valores foram interpolados por um valor médio das observações, calculado com base em cerca de 38 mil no intuito de uniformizar a série temporal.

Para a estrutura de dados foi utilizada a biblioteca [pandas[35]], onde os mesmos foram organizados em um [dataframe]⁷ com duas colunas. A coluna *date* que representa o dia, mês, ano e horário da observação e a coluna *sales* onde se encontra o valor de venda da moeda daquele determinado dia e horário.

⁷[Estrutura de dados bidimensional com os dados alinhados de forma tabular em linhas e colunas, mutável em tamanho e potencialmente heterogêneo.]

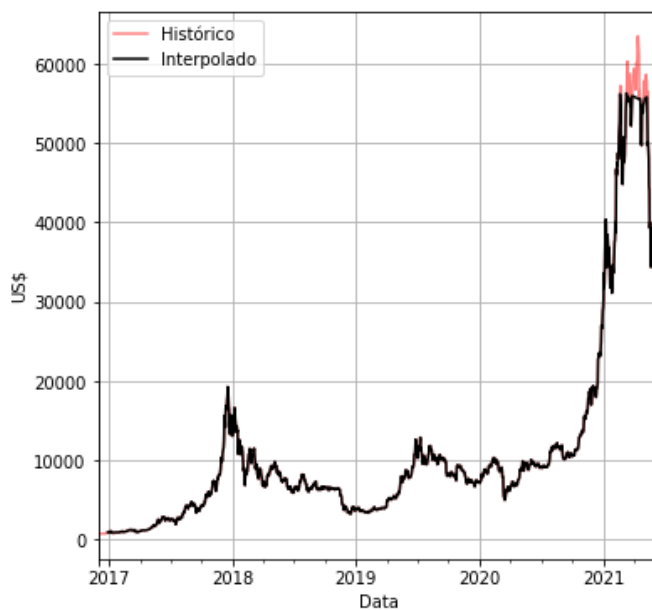


Fig. 9. Dados considerados outliers (em vermelho).

a.3. Máquina Virtual

Para o processamento e tratamento de todos os dados, foi utilizada uma máquina virtual com 16 CPUs (do inglês, *Central Processing Unit*) e 64GB de memória RAM (do inglês, *Random Access Memory*) da plataforma Google Cloud⁸. O sistema operacional utilizado foi o Linux Ubuntu 16.04 LTS (do inglês, *Long Term Support*), com a distribuição Anaconda⁹ e o ambiente Jupyter Notebook¹⁰ para construção e execução dos códigos.

a.4. Modelos

Como dito anteriormente, quatro modelos foram selecionados para realizar a previsão dos dados: regressão linear e polinomial, SARIMAX, LSTM e Prophet.

Os dados foram divididos em dois conjuntos: o primeiro conjunto consiste nos dados utilizados para treino, representados na Figura 10.

Já o segundo conjunto retrata os dados utilizados para teste, exibidos na Figura 11. A quantidade de dados selecionados para compor os conjuntos representam respectivamente, 99% e 1% da soma total das observações.

a.5. Previsão

Definiu-se o prazo de 1% dos dados processados convertidos em dias como o limite para a previsão dos modelos, onde os mesmos representam cerca de 16 dias. Em razão da alta volatilidade das criptomoedas, longos prazos de previsão comprometeriam a precisão dos modelos, de modo a prejudicar o resultado final.

⁸Suíte da Google para computação em nuvem. Disponível em <https://cloud.google.com/>

⁹Distribuição da linguagem Python para computação científica. Disponível em <https://www.anaconda.com/>.

¹⁰Plataforma de computação interativa de código aberto. Disponível em <https://jupyter.org/>.

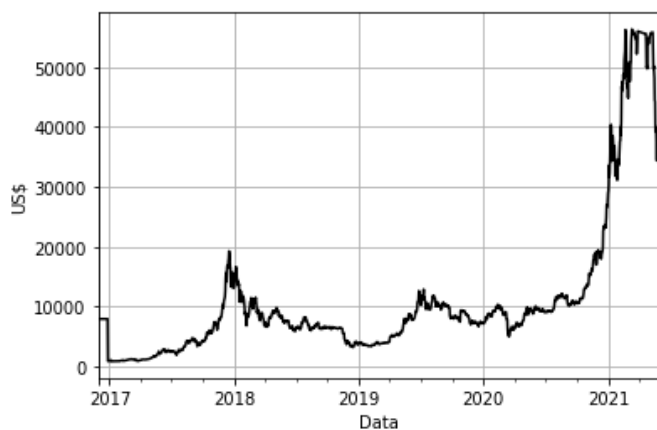


Fig. 10. Dados utilizados para treinamento dos modelos.

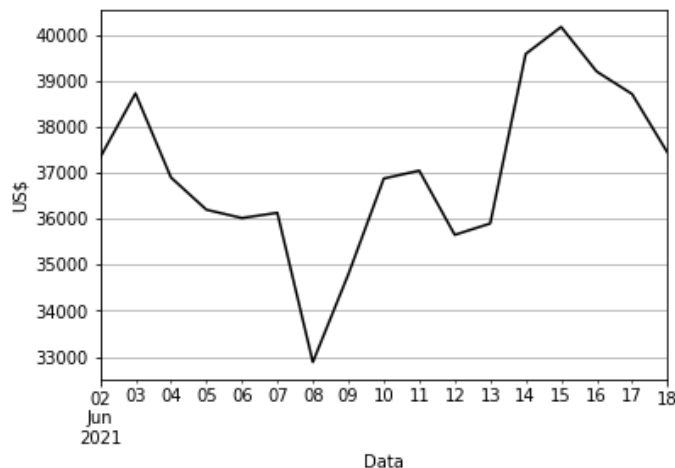


Fig. 11. Dados utilizados para testar e avaliar os modelos.

B. Resultados

Visando facilitar a análise e entendimento dos resultados, estes foram sintetizados em gráficos para cada modelo utilizado. Adicionalmente, as tabelas I, II e III trazem as métricas de forma detalhada de acordo com cada modelo.

b.1. Visualização das previsões

- Regressão linear: como pode ser visto no gráfico da Figura 12, este modelo aponta uma previsão baseada em um polinômio de primeiro grau (isto é, uma reta). Os resultados obtidos com os dados de teste podem ser observados na Figura 13.
- Regressão polinomial: exibida na Figura 14, este modelo estima o comportamento da série temporal através de uma função de grau 32 escolhida pelos autores. O grau do polinômio desta função pode ser alterado, porém ao custo de consumir mais memória e processamento. Ao se utilizar um valor acima de 96 para o grau do polinômio, o modelo é interrompido pois os valores calculados ultrapassam o valor máximo permitido para se armazenar em uma variável de ponto flutuante de 64 *bits*. Foram testados graus maiores que 32, porém, não houve mudança notável no valor das métricas e tempo de processamento. Os resultados obtidos com os dados de teste podem ser observados na Figura 15.

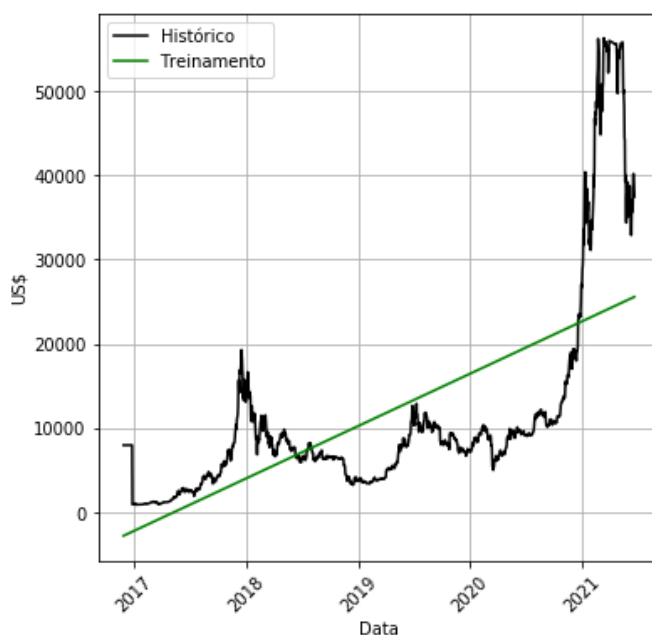


Fig. 12. Resultado do ajuste obtido pelo modelo de regressão linear sobre o conjunto de treinamento.

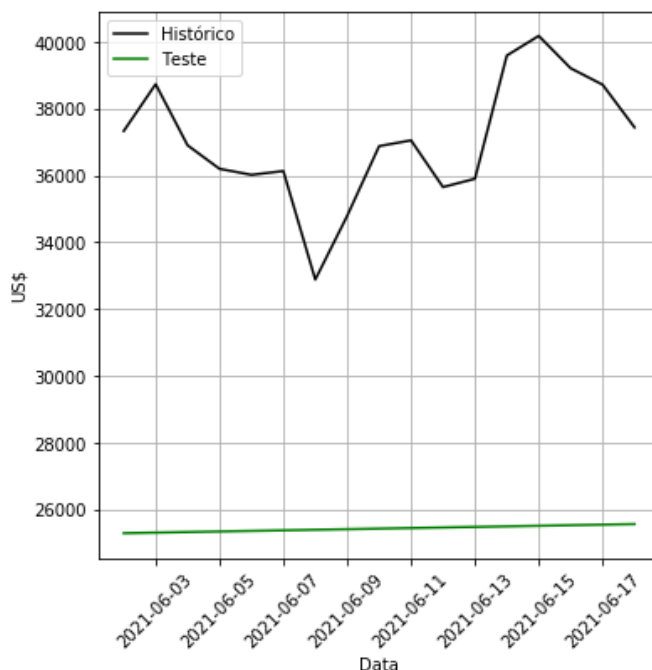


Fig. 13. Resultado do ajuste obtido pelo modelo de regressão linear sobre o conjunto de teste.

- SARIMAX: para este modelo, o resultado pode ser observado na Figura 16. Após os ajustes, o modelo foi treinado. Os resultados obtidos com os dados de teste podem ser observados na Figura 17.
- Prophet: necessita que os dados estejam em um formato padrão da biblioteca *fbprophet*, sendo então necessário realizar um pequeno ajuste no nome das colunas. Apesar disso, não houve alteração nos valores de treinamento e teste. Este modelo permite realizar uma previsão base-

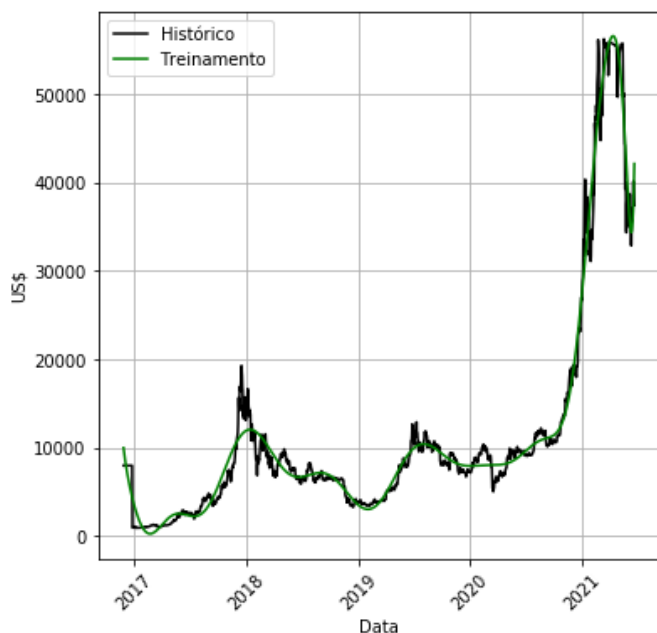


Fig. 14. Resultado do ajuste obtido pelo modelo de regressão polinomial sobre o conjunto de treinamento.

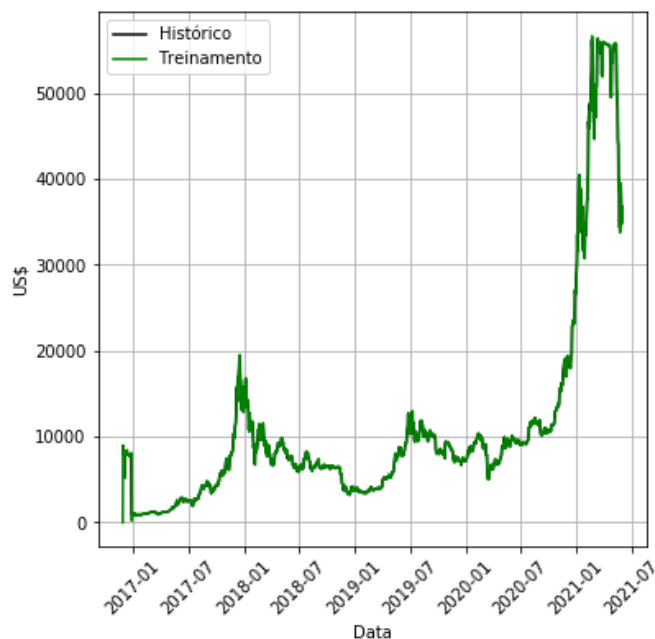


Fig. 16. Resultado do ajuste obtido pelo modelo SARIMAX sobre o conjunto de treinamento.

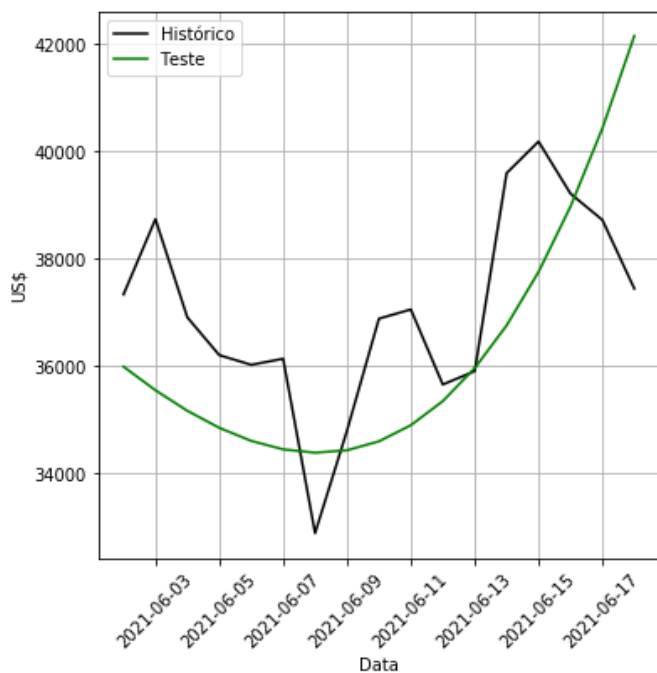


Fig. 15. Resultado do ajuste obtido pelo modelo de regressão polinomial sobre o conjunto de teste.

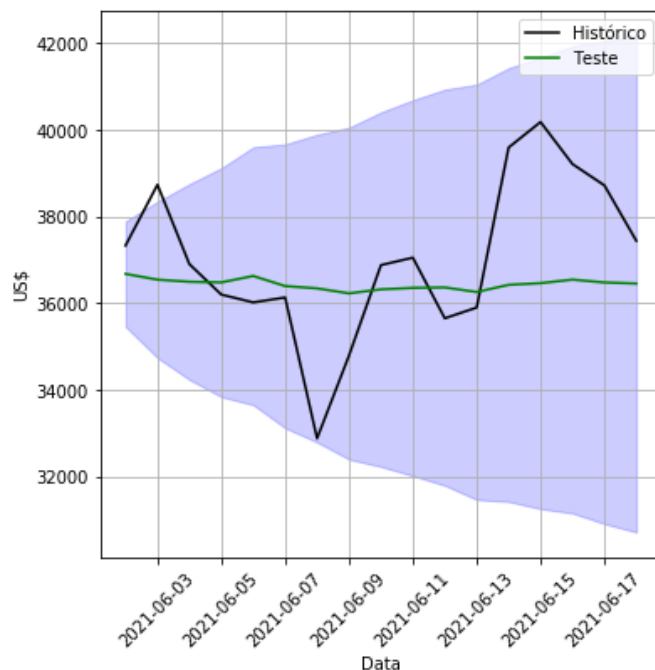


Fig. 17. Resultado do ajuste obtido pelo modelo SARIMAX sobre o conjunto de teste.

ada em feriados, entretanto para a análise de criptomoedas, isso foi desconsiderado. Após o modelo ser treinado, o resultado é descrito na Figura 18.

Como resultado do treinamento do modelo Prophet, a previsão é exibida na Figura 19.

- LSTM: para este modelo foi empregada uma dimensão 120×1 por conta do tempo de processamento e precisão. A dimensão escolhida é utilizada para estabelecer uma proporção, no qual 120 dias são utilizados para a base da previsão de 1 dia. Para a função de ativação foi selecionada

a ReLU (do inglês, *Rectified Linear Unit*), pois permite que a rede neural aprenda dependências não lineares. Isso torna o modelo mais fácil de treinar, de modo a atingir melhores resultados Brownlee [36]. Após treinar o modelo, os resultados estão dispostos na Figura 20.

O modelo não foi capaz de acompanhar a curva da série temporal. Portanto, seus resultados de teste não foram eficazes, conforme apresentado na Figura 21

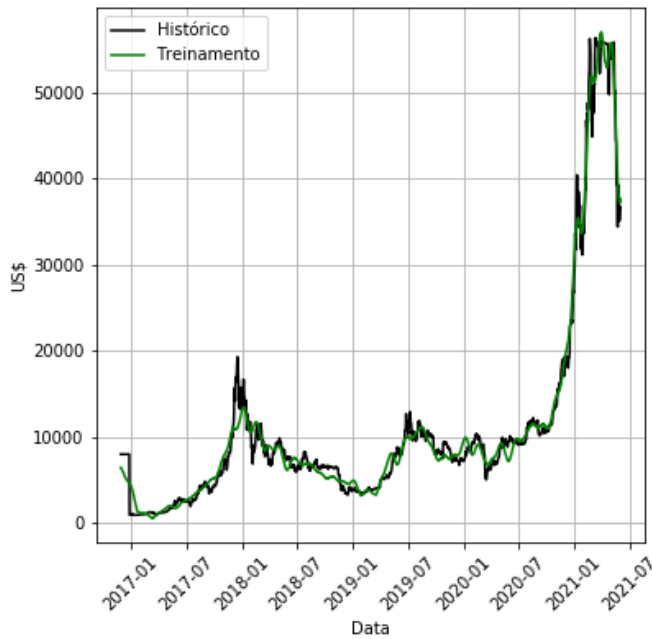


Fig. 18. Resultado do ajuste obtido pelo modelo Prophet sobre o conjunto de treinamento.

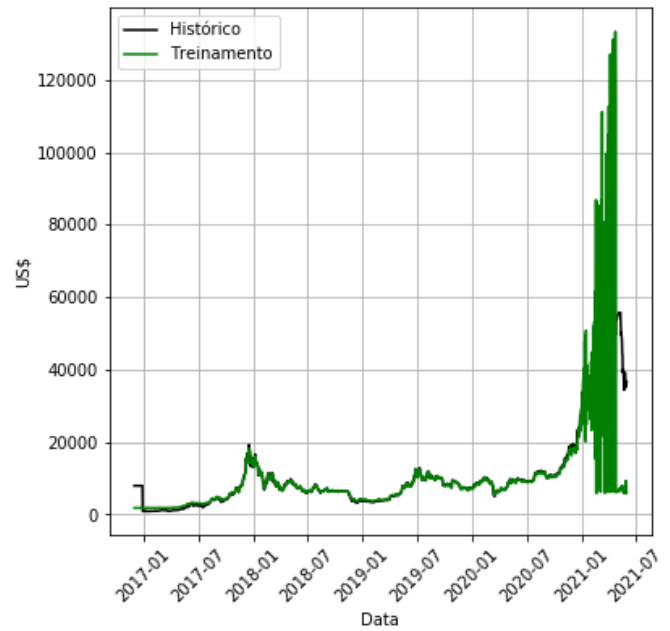


Fig. 20. Resultado do ajuste obtido pelo modelo LSTM sobre o conjunto de teste.

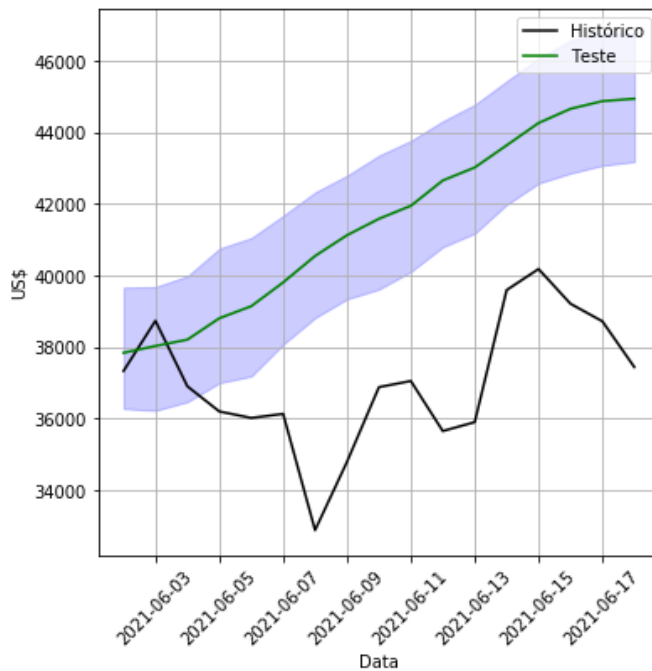


Fig. 19. Resultado do ajuste obtido pelo modelo Prophet sobre o conjunto de teste.

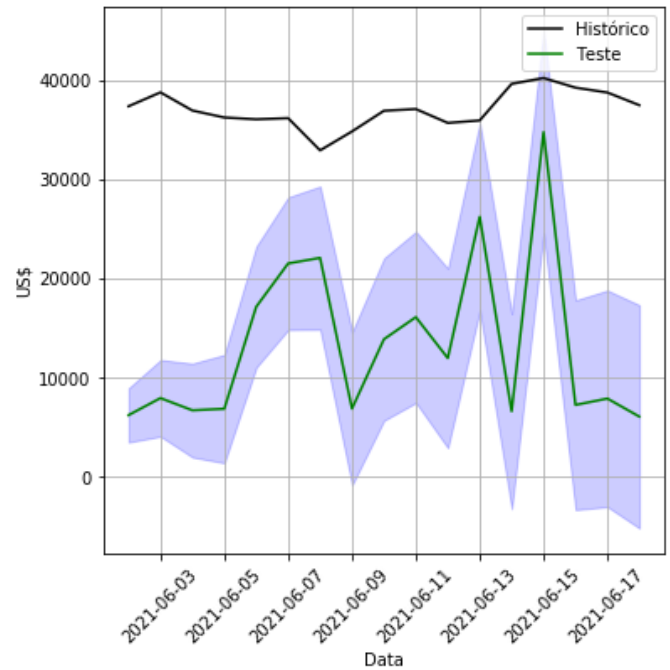


Fig. 21. Resultado do ajuste obtido pelo modelo LSTM sobre o conjunto de teste.

b.2. Métricas

O critério de comparação dos modelos foi definido através de métricas obtidas após o processamento dos modelos.

As métricas a serem comparados são: MAE, MAPE, MSE, RMSE e tempo de processamento para a previsão do modelo.

As Tabelas I e II mostram os resultados calculados para cada modelo considerando o conjunto de treinamento e teste respectivamente, enquanto a Tabela III exibe o tempo gasto no processamento.

C. Discussão

Em vista da insegurança por parte dos investidores em relação ao mercado de criptomoedas, o presente estudo teve como objetivo comparar modelos de previsão de séries temporais aplicados às cotações dessas.

Com base nos resultados apresentados na seção anterior, verificou-se que os modelos de regressão linear e polinomial obtiveram as primeiras colocações no *ranking* de tempo de resposta. Entretanto, com base na métrica MAPE do conjunto de testes, o modelo de Regressão Linear apresentou valores de

TABELA I
[MÉTRICAS DE AVALIAÇÃO SOBRE O CONJUNTO DE TREINAMENTO.]

Modelo	MAPE	MAE	MSE	RMSE
Regressão linear	86%	7003.81	92.10×10^6	9596.93
Regressão polinomial	17%	1074.57	2.53×10^6	1592.39
SARIMAX	3%	278.00	0.42×10^6	651.00
Prophet	16%	988.00	1.91×10^6	1384.00
LSTM	17%	3273.00	120.84×10^6	10993.00

TABELA II
[MÉTRICAS DE AVALIAÇÃO SOBRE O CONJUNTO DE TESTE.]

Modelo	MAPE	MAE	MSE	RMSE
Regressão linear	31.0%	11601.8	137.7×10^6	11736.4
Regressão polinomial	5.0%	1727.3	4.31×10^6	2076.4
SARIMAX	4.0%	1440.0	3.5×10^6	1859.0
Prophet	12.0%	4553.0	25.9×10^6	5085.0
LSTM	64.0%	18648.0	639.7×10^6	25293.0

TABELA III
[TEMPO DE PROCESSAMENTO.]

Modelo	Tempo de processamento
Regressão linear	<1 min
Regressão polinomial	<1 min
SARIMAX	Aprox. 3 min
Prophet	Aprox. 2 min
LSTM	Aprox. 9h

31%, taxa de erro relativamente alta, principalmente quando comparada aos demais modelos analisados, com exceção do LSTM. Este resultado justifica-se por se tratar de um polinômio de primeiro grau, ou seja, não é capaz de acompanhar com precisão os valores da série temporal. Diferentemente do anterior, o modelo de Regressão Polinomial obteve menores taxas de erro, com um MAPE de 5%. Quanto ao Prophet, apresentou resultados medianos, visto que atingiu um MAPE de 12% e ocupou a terceira colocação no *ranking* de tempo de resposta.

Diferentemente dos anteriores, o modelo SARIMAX se destacou pelo menor percentual de erros encontrados, com um MAPE de 4%. Apesar de ser o segundo modelo que mais consome tempo de processamento, este apresentou valores exponencialmente menores que o primeiro colocado no *ranking* de tempo de resposta, LSTM. Assim, quando comparado aos demais, o LSTM acaba se tornando inviável pelo alto custo computacional e tempo de processamento.

Os melhores resultados do SARIMAX frente ao LSTM, diferem dos encontrados na literatura. Segundo a revisão sistemática da literatura realizada por Sezer, Gudelek e Ozbayoglu [9], ao comparar o desempenho de modelos de DL com os de aprendizado de máquina tradicionais, verificou-se que na maioria dos estudos, os modelos de DL foram superiores. No presente estudo, isso pode ser explicado pelo fato de o LSTM

demandar mais processamento que os demais modelos. Com uma máquina dedicada, com maior processamento e memória, os valores de erros médios e percentuais poderiam se tornar aceitáveis para a previsão de cotações de criptomoedas.

O estudo de Lahmiri e Bekiros [28] considera a aplicação de DL altamente eficiente na avaliação da previsibilidade das moedas digitais. Assim como Lahmiri e Bekiros [28], Sezer, Gudelek e Ozbayoglu [9] concluíram que apesar de as previsões financeiras serem uma área com longa história de pesquisa, o surgimento de novos modelos de DL tem aumentado o interesse e oportunidade aos pesquisadores.

Entretanto, é importante mencionar que diferente das moedas convencionais, o valor das criptomoedas pode sofrer inúmeras influências externas como especulações em redes sociais e notícias de cunho econômico e social [7]. Nesse âmbito, destaca-se o trabalho de Li et al. [30], realizado a partir de uma análise dos sinais do Twitter como meio de sentimento do usuário para prever as flutuações de preço de uma criptomoeda. Segundo Pintelas et al. [2], o preço das criptomoedas segue um processo aleatório, com poucos padrões ocultos, o que dificulta sua identificação por estruturas inteligentes e impede a formação de previsões precisas e confiáveis.

A partir dos resultados demonstrados, o modelo SARIMAX se destaca dos demais pela sua precisão de previsão dos valores de criptomoedas, configurando-se como um meio promissor para a predição de séries temporais aplicados a esse mercado, de modo a aumentar a confiabilidade dos investidores. Contudo, em relação as várias influências externas sofridas pelas criptomoedas, ainda se faz necessária a obtenção de mais informações de entrada além dos dados utilizados neste estudo, como por exemplo, sentimentos baseados em redes sociais como o Twitter e dados de notícias.

VI. CONCLUSÃO

É evidente a rápida expansão do mercado de criptomoedas, tornando-se um dos investimentos com maior valorização mundial. Entretanto, as criptomoedas apresentam uma alta variabilidade de preço quando comparadas às moedas convencionais, tornando-se um ativo financeiro de risco. Em vista dessa volatilidade, investidores inexperientes deixam de investir com receio de tomarem decisões equivocadas.

Visando abordar tal problema, investidores vem demonstrando crescente interesse em técnicas de predição de séries temporais financeiras nos últimos anos. Dentre elas, destacam-se modelos de inteligência artificial, tanto de aprendizado de máquina quanto de aprendizado profundo.

Encontrar os melhores modelos capazes de identificar padrões de variação do valor das criptomoedas com precisão é de suma importância para auxiliar investidores a prever o melhor momento de comercialização dessas, aumentando a confiança neste setor e, potencialmente, seus ganhos.

Este trabalho utilizou modelos de regressão linear e polinomial, SARIMAX, Prophet e LSTM. O objetivo principal foi encontrar, após analisar a eficácia de cada modelo, o melhor para a previsão de séries temporais aplicadas a criptomoedas.

Com a análise dos resultados foi possível concluir que o modelo SARIMAX seria o mais indicado para tal tarefa. Isto porque apesar de os modelos de Regressão Linear e Polinomial

possuírem um rápido tempo de resposta, o primeiro atingiu altos percentuais de erro, enquanto o segundo obteve resultados inferiores ao SARIMAX. Já o modelo Prophet obteve resultados medianos em comparação com os demais. Por fim, o modelo LSTM apresentou alto custo de processamento, tornando-se inviável para a previsão de curto prazo.

Como sugestões para trabalhos futuros, são propostos os seguintes pontos:

- Considerar outros modelos de aprendizado de máquina: há diversos modelos com o mesmo propósito que podem ser avaliados, como aqueles baseados em DL;
- Levar em conta a influência de publicações em redes sociais e demais meios de comunicação: o sentimento do usuário e notícias em tais meios pode ser utilizado para prever as flutuações de preço de criptomoedas, visto que podem impactar na cotação do mercado;
- Considerar a cotação do dólar em determinados países;
- Considerar os horários de pico no mercado de criptomoedas;
- Avaliar quais empresas, estabelecimentos ou agentes financeiros aceitam criptomoedas como forma de pagamento e qual seu impacto no mercado financeiro, visto que suas decisões estratégicas pode influenciar na cotação;
- Considerar o crescimento da popularidade das criptomoedas e qual o impacto na cotação das mesmas, pois os volumes de negociação também podem influenciar nas cotações.

O projeto, que foi desenvolvido sob licença de código aberto, pode ser encontrado no seguinte repositório do GitHub: <https://github.com/samuelcsouza/cryptocurrency-forecasting>.

REFERÊNCIAS

- [1] Michael Nofer, Peter Gomber, Oliver Hinz e Dirk Schiereck. “Blockchain”. Em: *Business & Information Systems Engineering* 59.3 (2017), pp. 183–187.
- [2] Emmanuel Pintelas, Ioannis E. Livieris, Stavros Stavroyiannis, Theodore Kotsilieris e Panagiotis Pintelas. “Investigating the Problem of Cryptocurrency Price Prediction: A Deep Learning Approach”. Em: *Artificial Intelligence Applications and Innovations*. Ed. por Ilias Maglogiannis, Lazaros Iliadis e Elias Pimenidis. Cham: Springer International Publishing, 2020, pp. 99–110. ISBN: 978-3-030-49186-4.
- [3] Antonio Temóteo. *Bitcoin se valoriza 419% em reais em um ano; você deve investir em 2021?* 2021. URL: <https://economia.uol.com.br/noticias/redacao/2021/01/07/bitcoin-valorizacao-risco-perspectivas.htm> (acesso em 15/04/2021).
- [4] Blockchain. *O preço médio de mercado em dólares americanos nas principais bolsas de bitcoin*. 2021. URL: <https://www.blockchain.com/pt/charts/market-price> (acesso em 15/04/2021).
- [5] Decrypt. *Bitcoin em números: Uma retrospectiva de 2020*. 2020. URL: <https://portaldobitcoin.uol.com.br/bitcoin-em-numeros-uma-retrospectiva-de-2020/> (acesso em 15/04/2021).
- [6] Franco Valencia, Alfonso Gómez-Espinosa e Benjamín Valdés-Aguirre. “Price movement prediction of cryptocurrencies using sentiment analysis and machine learning”. Em: *Entropy* 21.6 (2019), p. 589.
- [7] Charllon Lobo Almeida. “Predição de séries temporais aplicada ao mercado de criptomoedas.” 2019.
- [8] Rodrigo Tolotti. *Coinbase estreia na Nasdaq com alta de 31% e valor da corretora de criptomoedas fica em US\$ 85,8 bi*. 2021. URL: <https://tinyurl.com/infomoney-tolotti-coinbase> (acesso em 15/04/2021).
- [9] Omer Berat Sezer, Mehmet Ugur Gudelek e Ahmet Murat Ozbayoglu. “Financial time series forecasting with deep learning: A systematic literature review: 2005–2019”. Em: *Applied Soft Computing* 90 (2020), p. 106181.
- [10] Raphael Campos. *Árvores de Decisão*. 2017. URL: <https://tinyurl.com/medium-campos-arvores> (acesso em 22/06/2021).
- [11] Gabriela Entringe. *Introdução à Regressão Linear Bayesiana*. 2019. URL: <https://tinyurl.com/medium-entringe-bayes> (acesso em 22/06/2021).
- [12] Ana Carolina Lorena e André CPLF de Carvalho. “Uma introdução às support vector machines”. Em: *Revista de Informática Teórica e Aplicada* 14.2 (2007), pp. 43–67.
- [13] André Ponce de Leon F. de Carvalho. *Redes Neurais Artificiais*. 2009. URL: <https://sites.icmc.usp.br/andre/research/neural/> (acesso em 22/06/2021).
- [14] Michael H Kutner, Christopher J Nachtsheim, John Neter, William Li et al. *Applied linear statistical models*. Vol. 5. McGraw-Hill Irwin Boston, 2005.
- [15] Ayush Pant. *Introduction to Linear Regression and Polynomial Regression*, Ayush Pant. 2019. URL: <https://tinyurl.com/towards-pant-regression> (acesso em 03/05/2021).
- [16] Rubens Cividati. “Comparação de Modelos de Previsão de Séries Temporais aplicados à Economia virtual de um MMORPG”. Em: *Anais do XXXIII Congresso De Iniciação Científica do Inatel* (2021).
- [17] Vasily Derbentsev, Natalia Datsenko, Olga Stepanenko e Vitaly Bezkorovainyi. “Forecasting cryptocurrency prices time series using machine learning approach”. Em: *SHS Web of Conferences*. Vol. 65. EDP Sciences. 2019, p. 02001.
- [18] Datascience George. *A Brief Introduction to ARIMA and SARIMAX Modeling in Python*. 2020. URL: <https://tinyurl.com/medium-george-arima> (acesso em 15/06/2021).

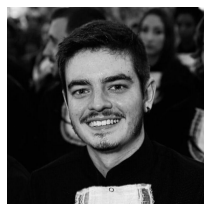
- [19] Sepp Hochreiter e Jürgen Schmidhuber. “Long short-term memory”. Em: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [20] Christopher Olah. *Understanding LSTM Networks*. 2015. URL: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (acesso em 25/04/2021).
- [21] Mauro Di Pietro. *Time Series Forecasting: ARIMA vs LSTM vs PROPHET*. 2020. URL: <https://tinyurl.com/medium-pietro-comparacao> (acesso em 14/05/2021).
- [22] Lucas Coutinho Vassali. “Aplicação de Redes Neurais LSTM para a Previsão de Curto Prazo de Vazão do Rio Paraíba do Sul”. 2018.
- [23] tensorflow. *tf.keras.layers.LSTM — TensorFlow Core*. 2021. URL: <https://www.tensorflow.org/api%5C%5Fdocs/python/tf/keras/layers/LSTM> (acesso em 17/05/2021).
- [24] Facebook. *Prophet - Forecasting at scale*. 2021. URL: <https://facebook.github.io/prophet/> (acesso em 25/04/2021).
- [25] R. Castelhão. “Utilização de Redes Neurais para Previsão no Mercado de Ações”. 2018.
- [26] sktime. *Prophet - sktime documentation*. 2020. URL: <https://tinyurl.com/sktime-docs-prophet> (acesso em 17/05/2021).
- [27] Vasily Derbentsev, Andriy Matviychuk e Vladimir N Soloviev. “Forecasting of Cryptocurrency Prices Using Machine Learning”. Em: *Advanced Studies of Financial Technologies and Cryptocurrency Markets*. Springer, 2020, pp. 211–231.
- [28] Salim Lahmiri e Stelios Bekiros. “Cryptocurrency forecasting with deep learning chaotic neural networks”. Em: *Chaos, Solitons & Fractals* 118 (2019), pp. 35–40.
- [29] Nor Azizah Hitam e Amelia Ritahani Ismail. “Comparative performance of machine learning algorithms for cryptocurrency forecasting”. Em: *Indonesian Journal of Electrical Engineering and Computer Science* 11.3 (2018), pp. 1121–1128.
- [30] Tianyu Ray Li, Anup S. Chamrajnagar, Xander R. Fong, Nicholas R. Rizik e Feng Fu. “Sentiment-Based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model”. Em: *Frontiers in Physics* 7 (2019), p. 98. ISSN: 2296-424X.
- [31] scikit-learn developers. *An introduction to machine learning with scikit-learn*. 2020. URL: <https://scikit-learn.org/stable/tutorial/basic/tutorial.html> (acesso em 12/07/2021).
- [32] pypi. *pmdarima*. 2021. URL: <https://pypi.org/project/pmdarima/> (acesso em 12/07/2021).
- [33] Romain Tavenard. *tslearn's documentation*. 2017. URL: <https://tslearn.readthedocs.io/en/stable/> (acesso em 12/07/2021).
- [34] SciPy community. *SciPy API*. 2021. URL: <https://docs.scipy.org/doc/scipy/reference/> (acesso em 12/07/2021).
- [35] The pandas development team. *Getting started*. 2021. URL: https://pandas.pydata.org/docs/getting_started/index.html (acesso em 12/07/2021).
- [36] Jason Brownlee. *A Gentle Introduction to the Rectified Linear Unit (ReLU)*. 2019. URL: <https://tinyurl.com/mastery-brownlee-relu> (acesso em 22/06/2021).

AUTORES

Felippe Feliciano de Carvalho é graduando em Engenharia da Computação pelo Instituto Nacional de Telecomunicações (Inatel) e estagiário em desenvolvimento de aplicações do Inatel Competence Center na área DWI, e possui interesse em backend e documentação.

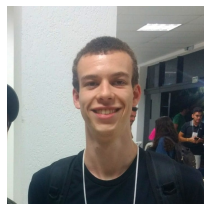


Guilherme Costa de Oliveira é técnico em telecomunicações com ênfase em automação pela escola técnica Francisco Moreira da Costa (ETE) e graduando em Engenharia de Computação pelo Instituto Nacional de Telecomunicações (Inatel). É desenvolvedor de aplicações na empresa TagoIO. Possui interesse em soluções



de problemas utilizando tecnologia, front-end e estilização de interface de usuário.

Samuel da Cruz Souza é graduando em Engenharia da Computação pelo Instituto Nacional de Telecomunicações (Inatel) e estagiário em desenvolvimento de aplicações Web na empresa 4Intelligence. Já autou como monitor da disciplina de Programação Orientada a Objetos e possui interesse na área de Ciência de Dados e Machine Learning.



Túlio Wilian Caproni é técnico em informática pelo Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais (Campus Machado) e graduando em Engenharia de Computação pelo Instituto Nacional de Telecomunicações (Inatel). Já atuou como monitor na disciplina de Projeto Integrado. Atualmente trabalha como desenvolvedor mobile na empresa Leucotron Telecom. Possui interesse na área



de desenvolvimento mobile e Web.



Marcelo Vinícius Cysneiros Aragão é graduado em Engenharia de Computação pelo Instituto Nacional de Telecomunicações (Inatel) em 2014 e Mestre em Ciência e Tecnologia da Computação pela Universidade Federal de Itajubá em 2018. Trabalhou de 2011 a 2018 no Inatel Competence Center, mais recentemente como Especialista em Sistemas, onde atuou principalmente como desenvolvedor de soluções de

Business Support Systems (BSS) em ambiente de integração contínua. É professor de disciplinas da graduação, como Inteligência Computacional e Redes Neurais, coordenador do curso de pós-graduação em Desenvolvimento de Aplicações para Dispositivos Móveis e *Cloud Computing* e doutorando em Engenharia de Telecomunicações no Inatel. Possui interesse nas áreas de aprendizado de máquina, ciência de dados e engenharia de *software*.