

Predição da cotação Real/Bitcoin usando a rede neural Long Short Term Memory

Real/Bitcoin price prediction using the Long Short Term Memory neural network

DOI:10.34117/bjdv9n6-159

Recebimento dos originais: 23/05/2023

Aceitação para publicação: 26/06/2023

Ramon Soares Mendes Meneses Leite

Bacharel em Ciência da Computação

Instituição: Departamento de Ciência da Computação - Universidade Federal de Catalão

Endereço: Av. Dr. Lamartine Pinto de Avelar, 1120, St. Universitário, Catalão – GO,

CEP: 75705-220

E-mail: ramonsoares@discente.ufcat.edu.br

Tércio Alberto dos Santos Filho

Doutor em Engenharia Elétrica

Instituição: Departamento de Ciência da Computação - Universidade Federal de Catalão

Endereço: Av. Dr. Lamartine Pinto de Avelar, 1120, St. Universitário, Catalão – GO,

CEP: 75705-220

E-mail: tercioas@ufcat.edu.br

Wanderlei Malaquias Pereira Junior

Doutor em Ciências Exatas e Tecnológicas

Instituição: Departamento de Ciência da Computação - Universidade Federal de Catalão

Endereço: Av. Dr. Lamartine Pinto de Avelar, 1120, St. Universitário, Catalão – GO,

CEP: 75705-220

E-mail: wanderlei.junior@ufcat.edu.br

Sérgio Francisco da Silva

Doutor em Ciência da Computação

Instituição: Departamento de Ciência da Computação - Universidade Federal de Catalão

Endereço: Av. Dr. Lamartine Pinto de Avelar, 1120, St. Universitário, Catalão – GO,

CEP: 75705-220

E-mail: sergio@ufcat.edu.br

RESUMO

Criptomoedas têm se tornado um ativo financeiro importante para empresas, governos e civis, sendo aceitas para transações financeiras em substituição às moedas tradicionais. Contudo as criptomoedas, que são atrativas por algumas questões como não ter controle governamental, apresentam alta volatilidade, sendo importante as predições de cotação de forma a ter uma base para decidir quando comprar e quando vender este tipo de ativo, de forma a maximizar lucros. Como a predição quando feita por pessoas demanda de especialistas, que têm altos custos e também são sujeitos a vieses e erros, vários estudos têm investigado a predição via técnicas computacionais de inteligência artificial. Neste estudo, investigou-se a aplicação de uma rede neural LSTM para a predição da cotação da criptomoeda Bitcoin em Reais, realizando a análise dos

parâmetros da rede neural, juntamente com a modelagem da rede para predição de um único valor para um dado horizonte de tempo futuro e para a predição de múltiplos valores para múltiplos instantes de tempo futuros. Os resultados obtidos indicam que o modelo de saída única tem uma boa performance para predição de curto prazo, principalmente na predição para o dia seguinte (horizonte 1) com um *mean absolute percentage error* (MAPE) de 2,57% na sua melhor configuração, contudo, esse erro não é linearmente escalável quando se aumenta o horizonte de predição. No modelo de múltiplas saídas produziu valores MAPE maiores para predições de curto prazo (horizonte 1, 3 e 5), contudo a taxa de aumento do erro é menor do que a do modelo de horizonte simples.

Palavras-chave: redes neurais, memórias de longo a curto prazo, bitcoin, predição de cotação.

ABSTRACT

Cryptocurrencies have become an important financial asset for companies, governments and civilians, being accepted for financial transactions in place of traditional currencies. However, cryptocurrencies, which are attractive for some reasons such as not having government control, have high volatility, making price predictions important in order to have a basis for deciding when to buy and when to sell this type of asset, in order to maximize profits. As prediction by people demands a specialist, which is expensive and also subject to biases and errors, several studies have investigated the application of artificial intelligence computational techniques for prediction. This study investigates the application of an LSTM neural network for predicting the price of the Bitcoin cryptocurrency in Reais, analyzing parameters of the neural network, together with the network modeling for the prediction of a single value for a given future time horizon and the prediction of multiple values for multiple future instants of time. The results obtained indicate that the single output model has a good performance for short-term loss, mainly in the prediction for the next day (horizon 1) with a mean absolute percentage error (MAPE) of 2.57% in its best configuration, however this error is not linearly scalable when the prediction horizon is increased. The multiple-output model produced higher MAPE values for short-term predictions (horizons 1, 3 and 5), however the error increase rate is lower than that of the simple horizon model.

Keywords: neural networks, long short term memory, bitcoin, quote prediction.

1 INTRODUÇÃO

De acordo com Nakamoto (2009), criptomoedas, também chamadas de moedas eletrônicas, são definidas por uma cadeia de assinaturas digitais (blockchain), onde um proprietário transfere a moeda para um destinatário, assinando digitalmente um hash da transação e informando a chave pública do próximo. Este mecanismo de operação financeira é altamente seguro e tem a vantagem de não depender de políticas governamentais e nem mesmo privadas, sendo um processo descentralizado. Conforme Hileman and Rauchs (2017), nos últimos anos, criptomoedas baseadas em blockchains,

introduzidas por Nakamoto (2009) com o Bitcoin, está crescendo de forma exponencial tanto em quantidade, quanto em volume de transações, influenciando os cenários comerciais globais e financeiros de forma significativa.

Sovbetov (2018) afirma que fatores relacionados ao mercado de criptomoedas, como beta de mercado, volume de negociação e volatilidade, parecem ser determinantes para criptomoedas como Bitcoin, Ethereum, Dash, Litecoin e Monero, tanto em curto prazo quanto em longo prazo. Os estudos do autor também mostram que existem muitos fatores que influenciam a tendência do preço da criptomoeda, o que a torna difícil de entender e prever. Assim, construir um mecanismo de previsão de cotação de criptomoedas se torna importante para potenciais investidores e agências governamentais, uma vez que poderia diminuir os riscos (evitando possíveis prejuízos), tanto em ações de compra quanto ações de venda.

Diante destes cenários, estudos como os de (AWOKE et al. 2020, BISWAS et al. 2021, INDULKAR 2021, WU et al. 2018) têm mostrado que é possível criar um modelo de previsão de séries temporais utilizando algoritmos de *deep learning*, como por exemplo, o *long short-term memory* (LSTM), que possui a característica de filtrar os dados relevantes ao longo do tempo, de forma a impedir os problemas de *vanishing and exploding gradients* e alcançar melhores resultados se comparado à redes neurais recorrentes (RNRs) tradicionais. Para o treinamento e teste do modelo, séries temporais de dados diários de criptomoedas podem ser coletadas através de *application programming interfaces* (APIs), como a API do CoinMarketCap, que é um site de monitoramento de preços de cripto ativos.

Neste trabalho é investigado o desempenho da rede neural LSTM para a predição da cotação Real/Bitcoin com bases em séries temporais multivariáveis. Foram feitos testes com modelos de saída única e saída múltipla, utilizando vários horizontes diferentes, além de variar parâmetros como o otimizador, tamanho de janela e tamanho de lote. Ao final dos experimentos, foi possível constatar que a rede LSTM pode gerar resultados satisfatórios para a predição da cotação do Bitcoin e que os modelos de saída única e saída múltipla têm comportamentos distintos conforme se varia o horizonte de predição.

2 LSTM

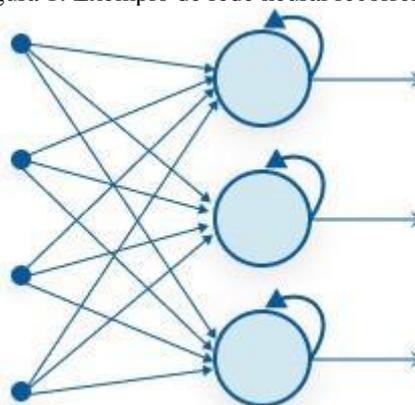
Conforme Haykin (2009), as redes neurais artificiais (RNAs) são modelos matemáticos inspirados no funcionamento dos neurônios biológicos. As RNAs com

múltiplas camadas de neurônios possuem a propriedade de suas funções de aproximação conseguirem mapear qualquer função não-linear sem qualquer estimativa antecedente relacionada aos dados. Devido suas características adaptativas, elas são eficazes no mapeamento da dinâmica de séries temporais não estacionárias e, nos últimos anos, têm colaborado para o desenvolvimento de métodos de previsão mais precisos para séries temporais (Kaushik and Giri 2020).

Existem também as Redes Neurais Recorrentes (RNR) que seguem o princípio das RNAs, porém, com uma arquitetura recorrente, isto é, possuem conexões recorrentes entre neurônios de processamento (Figura 1). Essa recorrência garante que as informações sequenciais sejam capturadas nos dados de entrada. Além disso, elas compartilham os parâmetros em diferentes etapas de tempo. Isso resulta em menos parâmetros para treinar, diminuindo o custo computacional.

Segundo Kaushik and Giri (2020), se tratando de previsão de séries temporais, a RNR mais usada é a *Long Short-Term Memory* (LSTM), que possui o diferencial de conseguir identificar a aleatoriedade dos dados melhor que uma RNA simples. A LSTM também é capaz de lidar com o erro de decaimento da retropropagação (problema presente em RNRs clássicas), devido a ter blocos de memória que auxiliam nisso, e isso a torna superior para previsão de séries temporais com longas dependências temporais.

Figura 1: Exemplo de rede neural recorrente.

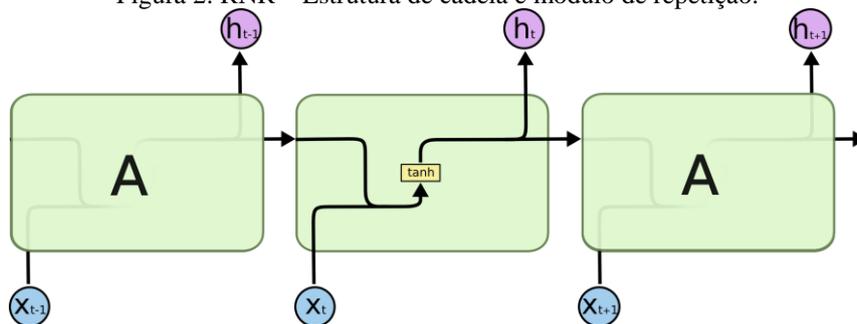


Fonte: Autores

As LSTMs foram propostas por Hochreiter and Schmidhuber (1997) especificamente para escapar do problema de dependência de longo prazo. A diferença entre as RNR tradicionais e LSTM está basicamente na estrutura. A RNR tradicional possui uma estrutura de ciclo, que pode ser visualizada como uma cadeia de módulos repetitivos ao longo do tempo (Figura 2). Considerando uma RNR padrão, as unidades

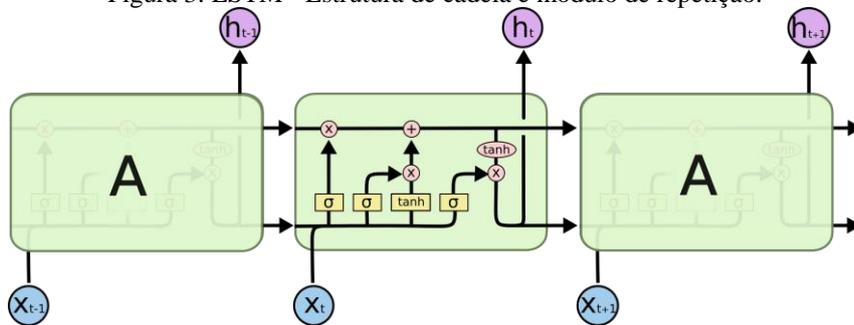
(neurônios) têm uma composição bastante elementar com uma simples função de ativação na saída destes, que alimenta ou retroalimenta neurônios deste módulo. Em uma LSTM, a estrutura de cadeia de informação temporal também existe, porém o módulo de repetição é bem diferente, pois a memória ao invés de ser implementada por ciclos de sinal podendo envolver múltiplos neurônios e camadas da rede, é implementada (retida e liberada) através de um conceito de portões (gates), que controla uma célula de memória (Figura 3).

Figura 2: RNR – Estrutura de cadeia e módulo de repetição.



Fonte: Autores

Figura 3: LSTM - Estrutura de cadeia e módulo de repetição.



Fonte: Autores

Uma célula LSTM, que é o elemento básico de uma rede LSTM, contém quatro elementos, sendo: três portões (*forget gate*, *input gate* e *output gate*) e o estado da célula (*cell state*), que tem o papel de guardar a memória que é manipulada.

O *forget gate* decide a quantidade de memória que vai permanecer e a quantidade que vai ser descartada. Ele analisa a entrada x_t e a saída do cálculo anterior h_{t-1} e utiliza uma função sigmóide para obter um valor entre 0 e 1. O valor 0 possui o significado de esquecer completamente, e o valor 1 significa lembrar de tudo. Este processo é relacionado com cada informação da memória, e após isso, o estado da memória é atualizado.

O *input gate* determina quais informações vão entrar na memória, analisa a

entrada x_t e a saída do cálculo anterior h_{t-1} , usa uma função sigmóide e uma função tangente hiperbólica (*tanh*) para normalizar os valores de entrada. O valor após passar pela função sigmóide é multiplicado pelo resultado da função *tanh*. De acordo com isso, o cálculo gera as informações que devem ser guardadas na memória. Por fim, se soma o valor final ao estado da célula para incrementar as informações na memória.

Por fim, o *Output gate* decide quais informações relevantes das etapas anteriores são necessárias para serem usadas como saída da célula. Ele analisa a entrada x_t e a saída do cálculo anterior h_{t-1} e aplica a função sigmóide. Após isso, o resultado é multiplicado pelo estado da célula aplicado em uma função *tanh*. Este processo resulta no processamento final do bloco de memória.

Estas células de memória LSTM podem ser organizadas de modo a ter múltiplas camadas e múltiplas células por camadas. Organizações complexas com muitas camadas são normalmente referidas como arquiteturas de *deep learning*.

3 TRABALHOS CORRELATOS

Nesta seção são descritas as principais pesquisas existentes na literatura que lidam com a predição de criptomoedas. Essas pesquisas são sumarizadas pela Tabela 1.

Conforme pode-se ver pela análise das pesquisas da Tabela 1, a ampla maioria das pesquisas (AWOKE et al. 2020, WU et al. 2018, IQBAL et al. 2021, INDULKAR 2021, DERBENTSEV et al. 2020, LI and DAI 2020, FLEISCHER et al. 2022) foca em predição da cotação de fechamento usando dados de periodicidade diária, fazendo a predição para o dia seguinte (horizonte 1). Uma exceção é a pesquisa de Jakubowicz and Abdelfattah (2021) que faz a predição da direção da cotação, se seja, se haverá valorização ou desvalorização, para a hora seguinte – neste caso o modelo é treinado com dados de periodicidade diária. Somente as pesquisas de (AWOKE et al. 2020) e (BISWAS et al. 2021) investigaram predições para um horizonte além de uma unidade. O principal motivo de muitas pesquisas focarem em predição somente para horizonte 1, é atribuído à dificuldade de fazer predições de longo prazo devido à alta volatilidade das criptomoedas.

Quanto aos modelos usados para as predições, os estudos apresentam uma apresentação uma predominância do uso de LSTM em conjunto com outro modelo, o que é normalmente referido como modelo híbrido. Como exemplos pode se citar os estudos de Biswas et al. (2021), que desenvolveu uma rede neural usando células LSTM e *gated recurrent unit* (GRU); Li and Dai (2020) que desenvolveu uma rede neural usando

camadas LSTM e convolucionais; Wu et al. (2018), que fez um modelo que combina LSTM com o modelo autoregressivo (AR); e Gao et al. (2021) que combina Heterocedasticidade Condicional Auto-Regressiva (GARCH) com LSTM. Em todos os estudos de modelos híbridos foram concluído que estes obtêm resultados superiores ao uso dos modelos base isoladamente, principalmente em cenários de alta volatilidade.

A principal criptomoeda analisada nos estudos é a Bitcoin, que é seguida pela criptomoeda Ethereum. A quantidade de instante de tempo, ou seja, comprimento das séries temporais usadas nos estudos variam de algumas centenas (209) à vários milhares (52608). A medida de desempenho mais usada conforme os estudos catalogados é o *root mean square error* (RMSE).

4 MATERIAL E MÉTODOS

Nessa seção será discutida a metodologia utilizada nesta pesquisa, desde obtenção dos dados da criptomoeda Bitcoin, os procedimentos de pré-processamento e treinamento de modelos utilizando redes LSTM.

4.1 BASE DE DADOS DE BITCOIN

A base de dados com informações históricas de transações do Bitcoin (BTC) utilizando a moeda real (BRL) foi obtida através do site Cryptodatadownload 2, que contém séries históricas de várias criptomoedas com diversas periodicidades e intervalos de tempo. A série temporal obtida é de periodicidade diária, datada entre 27-12-2020 e 08-08-2022, e possui as seguintes variáveis/colunas: unix, date, symbol, open, high, low, close, Volume BTC, Volume BRL e tradecount.

Os dados utilizados para este estudo consiste das variáveis open (prec,o de abertura), high (máxima do dia), low (mínima do dia), close (fechamento do dia), Volume BTC e Volume BRL, que representam o volume das transações em Bitcoin e em real, respectivamente.

4.2 PRÉ-PROCESSAMENTO DAS SÉRIES TEMPORAIS

Segundo Batista (2003), como os dados são o meio utilizado para a extração de conhecimento dentro do aprendizado de máquina, a qualidade e o pré-processamento adequado destes implica diretamente no desempenho que um modelo vai ter.

Há diversos fatores presentes em um determinado conjunto de dados que podem prejudicar o desempenho de uma rede neural. No caso dos dados de Bitcoin usados, não

há problemas como dados faltantes nas séries temporais, porém se tem diferenças grandes na ordem de grandeza das variáveis que compõem os dados, e isso pode ser resolvido através da normalização dos dados.

Tabela 1: Trabalhos correlatos

Autor(es)	Técnicas principais	Criptomoedas	Aplicação	Instantes de tempo / periodicidade	Horizontes de predição	Métodos de avaliação
Awoke et al. (2020)	LSTM e GRU	Bitcoin	Predição da cotação de fechamento	1512 / diária	1, 3, 5, 7 e 15	MAPE, RMSE
Wu et al. (2018)	LSTM convencional e híbrida	Bitcoin	Predição da cotação de fechamento	209 / diária	1	MSE, RMSE, MAE
Iqbal et al. (2021)	ARIMA, FBProphet e XG Boosting	Bitcoin	Predição da cotação de fechamento	2192 / diária	-	MAE, RMSE, R ²
Indulkar (2021)	LSTM	Bitcoin, Ethereum, Chainlink, Bitcoin Cash e XRP	Predição da cotação de fechamento	1095 / diária	-	MAE
Biswas et al. (2021)	GRU e LSTM	Litecoin e Monero	Predição da cotação de fechamento	1279 (Litecoin), 1851 (Monero) / diária	1, 3 e 7	RMSE
Derbentsev et al. (2020)	Random Forest e Gradient Boosting Machine	Bitcoin, Ethereum e Ripple	Predição da cotação de fechamento	2192 (Bitcoin e Ripple), 1974 (Ethereum) / diária	1	MAPE e RMSE
Gao et al. (2021)	GARCH e LSTM	Bitcoin	Predição da cotação de fechamento	2437 / diária	-	MSE, MAE e RMSE
Li and Dai (2020)	LSTM e CNN	Bitcoin	Predição da cotação de fechamento	608 / diária	1	MAE, RMSE, MAPE
Jakubowicz and Abdelfattah (2021)	Regressão Logística, Support Vector Machin, Random Forest, KNN e Decision Tree	Bitcoin	Predição da direção (valorização/desvalorização)	52608 / horária	1	Precisão
Fleischer et al. (2022)	LSTM	lectro-Optical System (EOS), Bitcoin, Ethereum e Dogecoin	Predição da cotação de fechamento	-	1	RMSE

Fonte: Autores

4.2.1 Normalização

Batista (2003) define normalização como a transformação dos valores originais de um conjunto de dados para valores em um determinado intervalo, normalmente entre -1 e 1 ou 0 e 1. Tal transformação possibilita que a diferença de escala entre as múltiplas variáveis que compõem as séries temporais não afetem a técnica de predição utilizada. No experimento foi utilizada a normalização de cada variável no intervalo entre 0 e 1, através das seguintes fórmulas aplicadas sequencialmente:

$$std = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (1)$$

$$scaled = std * (lim_{sup} - lim_{inf}) + lim_{inf} \quad (2)$$

Onde

X é um vetor de valores correspondentes a uma variável, $\min(X)$ e $\max(X)$ são respectivamente o menor e o maior valor de X , e lim_{sup} e lim_{inf} são os valores limites para a normalização, com lim_{inf} sendo o menor valor e lim_{sup} o maior valor. Nos experimentos lim_{inf} foi setado com o valor 0 e lim_{sup} foi setado com o valor 1.

4.3 ARQUITETURA DE LSTM PARA A PREDIÇÃO DE SÉRIES TEMPORAIS

Para este trabalho foi elaborado um modelo LSTM de múltiplas camadas (Figura 4), sendo uma camada de 1024 neurônios e uma de 512. Após as duas camadas LSTM é aplicada uma única camada densa, com a quantidade de neurônios igual a quantidade de dados a serem previstos (a quantidade de dados a serem previstos varia de experimento para experimento). A quantidade de neurônios da camada de entrada (que não fazem processamento) também varia conforme o experimento, sendo dada em função do tamanho de janela temporal utilizada, multiplicado pela quantidade de variáveis (seis variáveis, conforme a base de dados utilizada).

A função de ativação escolhida para a saída da rede neural de cada modelo foi a *linear*. A *tanh* foi utilizada como função de ativação das camadas LSTM.

Por fim, na saída da rede, o valor resultante é desnormalizado antes do cálculo de erro, sendo comparado com o valor real de cotação do conjunto de teste. Para a desnormalização usa-se a seguinte equação:

$$out' = out * (\max(Y) - \min(Y)) + \min(Y) \quad (3)$$

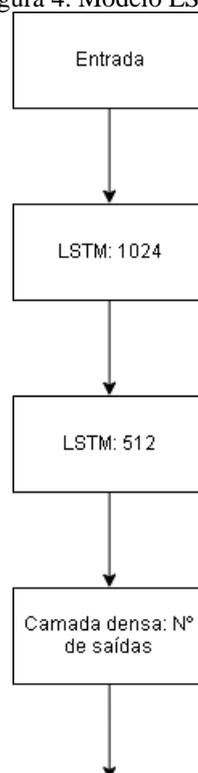
Onde

out é a predição normalizada gerada pela rede neural, $max(Y)$ e $min(Y)$, são respectivamente os valores máximo e mínimo da variável alvo da predição (no caso a cotação BRL/BTC) sobre o conjunto de treinamento. Assim, out' é a predição desnormalizada que é comparada com o respectivo valor real da cotação dada no conjunto de teste, para a mensuração do erro do modelo.

5 RESULTADOS

Foram feitos três experimentos utilizando a arquitetura LSTM especificada na Seção 4.3. O primeiro deles visa a experimentação de parâmetros de forma a decidir a parametrização mais promissora do modelo para os demais experimentos. Para isso, o modelo é configurado para prever uma saída única, que se trata da predição da cotação BRL/BTC para o dia seguinte. O segundo experimento visa analisar o desempenho deste modelo de única saída, quando se aumenta o horizonte de predição para 3, 5 e 7 dias. O terceiro experimento consiste da alteração do modelo para prever em uma única execução, múltiplas saídas, correspondentes à cotação BRL/BTC para os horizontes de predição (1, 3, 5 e 7 dias); neste caso o modelo passa a ter quatro neurônios na última camada.

Figura 4. Modelo LSTM.



Em todos os experimentos a base de dados de Bitcoin foi separada de forma sequencial conforme o tempo (data) em treinamento e teste, sendo 80% dos registros

iniciais usados para treinamento e 20%, que correspondem aos registros restantes usados para texto. Também, em todos os experimentos foram utilizadas 50 épocas de treinamento, que foram executadas utilizando uma taxa de aprendizado de valor 0,001. O objetivo sempre é conseguir prever o preço de fechamento do Bicoín, ou seja a cotação BRL/BTC no final do dia.

5.1 PARAMETRIZAÇÃO DO MODELO LSTM

O primeiro experimento teve como objetivo definir os parâmetros mais promissores do modelo. Para isso foram experimentados os valores 16, 32 e 64 para tamanho de lote (*batch*) de treinamento; 7, 14 e 28 para tamanho de janela deslizante; e *rmsprop* e *adam* como otimizadores. Ambos os otimizadores são baseados no cálculo de gradiente descendente.

O *rmsprop* (HINTON e SWERSKY, 2012) faz a atualização dos pesos usando uma suavização do tipo média móvel para calcular o termo v_n em função dos gradientes anteriores:

$$v_n = \gamma v_{n-1} + (1 - \gamma) \Delta_n^\omega \quad (4)$$

Onde

γ é um parâmetro que controla o peso do gradiente atual em relação aos gradientes anteriores. A regra de atualização dos pesos se torna:

$$\omega^{(l)}(n+1) = \omega^{(l)}(n) + \frac{\eta}{\sqrt{v_n + \varepsilon}} \Delta_n^\omega \quad (5)$$

Onde

η é a taxa de aprendizagem, Δ_n^ω corresponde o valor de gradiente, ε é um pequeno valor constante, para evitar divisão por zero. Para γ foi utilizado o valor 0.9 e para ε o valor 10^{-7} .

O otimizador adam, definido em Diederik P. Kingma (2014), mantém controle dois vetores de atualização com base no gradiente:

$$m_n = \beta_1 m_{n-1} + (1 - \beta_1) \Delta_n^\omega, \quad v_n = \beta_2 v_n + (1 - \beta_2) (\Delta_n^\omega)^2 \quad (6)$$

Onde

β_1 e β_2 são parâmetros que controlam a atuação dos vetores em função do gradiente momentâneo. Para evitar problemas numéricos esses vetores são normalizados da seguinte forma:

$$\hat{m}_n = \frac{m_n}{1 - \beta_1}, \quad \hat{v}_n = \frac{v_n}{1 - \beta_2} \quad (7)$$

A equação de atualização dos pesos de *adam* é dada por:

$$\omega_{n+1} = \omega_{n-1} - \frac{\eta}{\sqrt{\hat{v}_n + \epsilon}} \hat{m}_n \quad (8)$$

Onde

η é uma taxa de aprendizagem e ϵ é um fator para evitar divisão por zero. Nos experimentos com o otimizador *adam* foram utilizados os seguintes valores de parâmetros: $\beta_1 = 0.9$, $\beta_2 = 0.999$, e $\epsilon = 10^{-7}$.

Na Tabela 2 é mostrado o erro médio absoluto (MAE) e o erro percentual médio absoluto (MAPE) para cada uma das configurações, ordenado do menor ao maior MAE. De acordo com os testes realizados, vê-se que a melhor configuração foi o modelo com tamanho de *batch* igual a 16, otimizador *adam* e tamanho de janela igual a 7, que alcançou um MAPE de aproximadamente 2,57%. Em média, pode se notar que os modelos com o otimizador *adam* produziram menor taxa de erro que os modelos com *RMSProp*. Também se nota que tamanho de lote igual a 16 se destacou entre os tamanho testados. Em relação ao tamanho da janela, é observado que não há um padrão bem definido, porém os dois melhores modelos obtidos foram com tamanho de janela igual a 7. Adicionalmente, quanto menor o tamanho da janela, significa que o modelo gera a predição usando menos dados, pois a quantidade dados de entrada da rede é dada pelo tamanho da janela multiplicado pelo número que variáveis (este último fixo).

Tabela 2. Resultados dos modelos LSTM com saída única para horizonte 1.

Batch	Otimizador	Tamanho de janela	MAE	MAPE
16	adam	7	3551,57	0,02579
32	adam	7	3670,49	0,02704
16	adam	14	3697,02	0,02705
16	adam	28	3838,27	0,02836

16	rmsprop	28	3961,27	0,02888
64	adam	7	3962,11	0,02947
32	adam	14	3966,62	0,02970
32	adam	28	4313,80	0,03237
64	adam	14	4364,34	0,03283
32	rmsprop	28	4916,06	0,03642
64	adam	28	4954,75	0,03487
16	rmsprop	14	5119,51	0,03748
64	rmsprop	14	5253,52	0,03905
64	rmsprop	7	5765,83	0,04168
16	rmsprop	7	5873,01	0,04358
64	rmsprop	28	7053,00	0,05051
32	rmsprop	7	7481,14	0,05560
32	rmsprop	14	8031,63	0,05888

Fonte: Autores

5.2 MODELO COM SAÍDA ÚNICA

Seguindo a configuração de parâmetros de menor taxa de erro do experimento anterior (otimizador *adam*, *batch* 16 e tamanho de janela de 7 dias), foram feitos testes para os horizontes de 3, 5 e 7 dias, usando o modelo de saída única. Neste caso, o modelo é treinado independentemente para cada um dos horizontes. Os resultados obtidos são mostrados na Tabela 3. Percebe-se que os resultados possuem um comportamento explosivo (exponencial) de aumento do erro, pois a cada aumento de dois dias no horizonte, o valor do erro duplica em relação ao erro do horizonte anterior. Percebe-se que este modelo tende a gerar taxas de erros bastante altas para predições de longo prazo, por exemplo se fosse usado um horizonte de 15 dias. Assim, apesar da boa performance deste modelo para predição de curto prazo, por exemplo para o próximo dia, onde este tem um MAPE de 2.57%, este modelo é impreciso para predições de longo prazo.

Tabela 3. Resultados do modelo LSTM com saída única para horizontes 3, 5 e 7.

Horizonte	MAE	MAPE
1	3551,57	0,0257
3	6519,59	0,0496

5	10276,05	0,0827
7	21302,90	0,1777

Fonte: Autores

5.3 MODELO COM SAÍDA MÚLTIPLA

Para o treinamento desse modelo foi utilizada a mesma configuração obtida como melhor para o modelo de saída única (tamanho de batch 16, Adam como otimizador e tamanho de janela 7) usada no modelo anterior, porém, trocando a última camada para rede de um para quatro neurônios, para prever múltiplas saídas simultaneamente, com os horizontes 1, 3, 5 e 7, respectivamente. A Tabela 4 mostra os resultados do modelo de saída múltipla para os diferentes horizontes. Comparando os resultados deles com os resultados do modelo de saída única, percebe-se que apenas para o horizonte mais longo (7 dias) o modelo de múltiplas saídas se saiu melhor. Contudo, mesmo com a maioria dos valores de erro sendo maiores que os do modelo de saída única, parece-se que o modelo de saída múltipla teve um fator de crescimento de erro menor do que o modelo de saída única, quando se aumenta o horizonte.

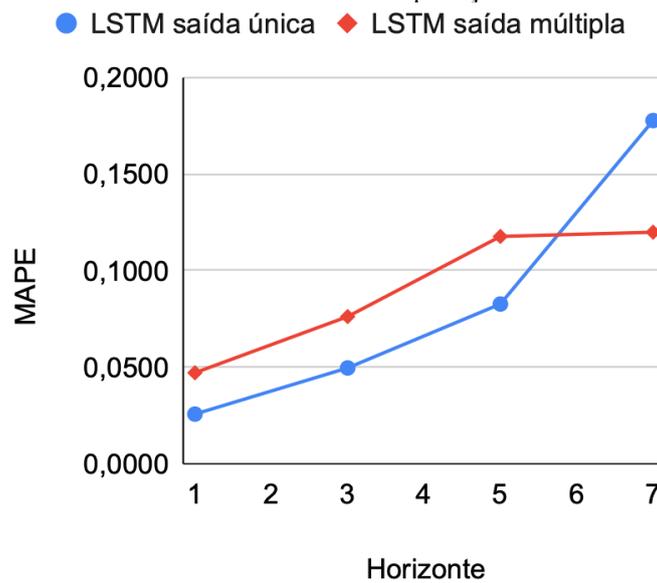
Tabela 4. Resultados modelo LSTM com saída múltipla.

Horizonte	MAE	MAPE
1	6358,02	0,0471
3	9833,59	0,0762
5	14445,76	0,1177
7	14814,25	0,1199

Fonte: Autores

Visando facilitar a análise dos modelos de horizonte único e do modelo de horizonte múltiplo quanto ao crescimento da taxa de erro MAPE, em função do horizonte de predição foi plotada a Figura 5. Contudo, pode se notar que no geral a taxa aumento do modelo de horizonte múltiplo é menor, sugerindo que este terá melhor desempenho que o modelo de saída simples a medida que o horizonte for aumentado além de 7 dias.

Figura 5. Taxas de erro MAPE do modelo LSTM com saída única e com saída múltiplas para os diferentes horizontes de predição.



Fonte: Autores

6 CONCLUSÕES

Dados os resultados apresentados, é possível observar que a arquitetura LSTM gerou uma variação considerável do desempenho conforme as configurações experimentadas. O modelo que se destacou dentre todos foi o modelo de saída única para horizonte 1, utilizando tamanho de lote 16, tamanho de janela 7 e o otimizador adam, que resultou em um MAPE de 2,57%. Contudo, verificou-se que o modelo de saída única tende a gerar taxas de erros muito altas para predições de longo prazo. Já o modelo de múltiplas saídas apresenta taxas de erros maiores para valores de horizonte pequenos (1, 3 e 5 dias) mas tem um fator de crescimento de erro menor quando aumenta-se o horizonte. Assim, se for considerado predição de longo prazo, para horizonte de 7 dias ou maiores, o modelo de saídas múltiplas é mais promissor.

REFERÊNCIAS

Awoke et al. 2020 Awoke, T., Rout, M., Mohanty, L., and Satapathy, S. C. (2020). Bitcoin price prediction and analysis using deep learning models. In Communication Software and Networks, pages 631–640. Springer Singapore.

Batista 2003 Batista, G. E. d. A. P. A. (2003). Pré-processamento de dados em aprendizado de máquina supervisionado. PhD thesis, Universidade de São Paulo, São Carlos : Instituto de Ciências Matemáticas e de Computação.

Biswas et al. 2021 Biswas, S., Pawar, M., Badole, S., Galande, N., and Rathod, S. (2021). Cryptocurrency price prediction using neural networks and deep learning. In 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE.

Derbentsev et al. 2020 Derbentsev, V., Datsenko, N., Babenko, V., Pushko, O., and Pursky, O. (2020). Forecasting cryptocurrency prices using ensembles-based machine learning approach. In 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T). IEEE.

Diederik P. Kingma 2014 Diederik P. Kingma, J. B. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980.

Fleischer et al. 2022 Fleischer, J. P., von Laszewski, G., Theran, C., and Bautista, Y. J. P. (2022). Time series analysis of cryptocurrency prices using long short-term memory. Algorithms, 15(7):230.

Gao et al. 2021 Gao, Z., He, Y., and Kuruoglu, E. E. (2021). A hybrid model integrating LSTM and garch for bitcoin price prediction. In 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP). IEEE.

Haykin 2009 Haykin, S. S. (2009). Neural networks and learning machines. Pearson Education, Upper Saddle River, NJ, third edition.

Hileman and Rauchs 2017 Hileman, G. and Rauchs, M. (2017). 2017 global cryptocurrency benchmarking study. SSRN Electronic Journal.

Hinton and Swersky 2012 Hinton, G., S. N. and Swersky, K. (2012). Neural networks for machine learning. lecture 6e. University of Toronto.

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

Indulkar 2021 Indulkar, Y. (2021). Time series analysis of cryptocurrencies using deep learning & fbprophet. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE.

Iqbal et al. 2021 Iqbal, M., Iqbal, M., Jaskani, F., Iqbal, K., and Hassan, A. (2021). Time-series prediction of cryptocurrency market using machine learning techniques. EAI Endorsed Transactions on Creative Technologies, 8(28):170286.

Jakubowicz and Abdelfattah 2021 Jakubowicz, E. and Abdelfattah, E. (2021). The rise

and fall of bitcoin: Predicting market direction using machine learning models. In 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE.

Kaushik and Giri 2020 Kaushik, M. and Giri, A. K. (2020). Forecasting foreign exchange rate: A multivariate comparative analysis between traditional econometric, contemporary machine learning amp; deep learning techniques.

Li and Dai 2020 Li, Y. and Dai, W. (2020). Bitcoin price forecasting method based on CNN-LSTM hybrid neural network model. J. Eng. (Stevenage), 2020(13):344–347.

Nakamoto 2009 Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system.

Sovbetov 2018 Sovbetov, Y. (2018). Factors influencing cryptocurrency prices: Evidence from bitcoin, ethereum, dash, litcoin, and monero. Journal of Economics and Financial Analysis, 2(2):1–27.

Wu et al. 2018 Wu, C.-H., Lu, C.-C., Ma, Y.-F., and Lu, R.-S. (2018). A new forecasting framework for bitcoin price with LSTM. In 2018 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE.