



UNIVERSIDADE FEDERAL DE GOIÁS  
INSTITUTO DE INFORMÁTICA

JOÃO GABRIEL SILVA FERNANDES

# **Previsão Generalizada à curto prazo de Criptomoedas com Redes Neurais Profundas**

Goiânia  
2021

UNIVERSIDADE FEDERAL DE GOIÁS  
INSTITUTO DE INFORMÁTICA

**AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHO DE  
CONCLUSÃO DE CURSO EM FORMATO ELETRÔNICO**

Na qualidade de titular dos direitos de autor, **AUTORIZO** o Instituto de Informática da Universidade Federal de Goiás – UFG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da UFG, entendendo-se os termos “reproduzir” e “publicar” conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela Universidade, a partir desta data.

**Título:** Previsão Generalizada à curto prazo de Criptomoedas com Redes Neurais Profundas

**Autor(a):** João Gabriel Silva Fernandes

Goiânia, 14 de Junho de 2021.

---

João Gabriel Silva Fernandes – Autor

---

<Nome do Orientador> – Orientador

---

<Nome do Co-orientador> – Co-Orientador

JOÃO GABRIEL SILVA FERNANDES

# **Previsão Generalizada à curto prazo de Criptomoedas com Redes Neurais Profundas**

Trabalho de Conclusão apresentado à Coordenação do Curso de <Nome do Programa de Pós-Graduação> do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Bacharel em <Nome do Programa de Pós-Graduação>.

**Área de concentração:** <Área de Concentração>.

**Orientador:** Prof. <Nome do Orientador>

**Co-Orientador:** Prof. <Nome do Co-orientador>

Goiânia  
2021

JOÃO GABRIEL SILVA FERNANDES

# Previsão Generalizada à curto prazo de Criptomoedas com Redes Neurais Profundas

Trabalho de Conclusão apresentado à Coordenação do Curso de <Nome do Programa de Pós-Graduação> do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Bacharel em <Nome do Programa de Pós-Graduação>, aprovada em 14 de Junho de 2021, pela Banca Examinadora constituída pelos professores:

---

**Prof. <Nome do Orientador>**

Instituto de Informática – UFG  
Presidente da Banca

---

**Prof. <Nome do Co-orientador>**

<Nome da Unidade Acadêmica do Co-orientador> – <Sigla da Universidade do Co-orientador>

---

**Prof. <Nome do membro da banca>**

<Unidade acadêmica> – <Sigla da universidade>

---

**Profa. <Nome do membro da banca>**

<Unidade acadêmica> – <Sigla da universidade>

---

## Resumo

---

S. Fernandes, João Gabriel. **Previsão Generalizada à curto prazo de Criptomoedas com Redes Neurais Profundas**. Goiânia, 2021. 22p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

Nos últimos anos o mercado de criptomoedas vem ganhando espaço no radar de investidores de alto risco, e consequentemente o interesse em prever movimentos deste mercado aumentou, problema no qual ainda se mostra desafiante. Esse trabalho faz o uso de redes neurais recorrentes **LSTM**, num modelo de regressão, para predição da variação diária de preço de Bitcoin e Altcoins. No lugar da previsão do preço, o alvo escolhido é sua variação (%). Validação cruzada K-Fold é realizada para uma medida mais confiável do modelo. Uma análise dos resultados é feita para a adição de dados sociais como volume de tweets, google trends e flags de eventos, ao final comparando com outros trabalhos. Dados de preço do mercado foram obtidos no *Kaggle*, enquanto volume de tweets, google-trends e eventos foram extraídos do *bitinfochart.com*, *trends.google.com* e *coinmarketcal.com*. Duas criptomoedas estáveis (BTC, ETH) e duas *Altcoins* (TRX, EOS) são usadas. Os resultados obtidos mostram um **MSE** de  $3,9 \cdot 10^{-4}$  e acurácia de *movimento-negativo-ou-positivo-no-preço* de 75,4% para Bitcoin, e performance comparativa para treino e teste cruzado entre moedas.

### Palavras-chave

<Palavra chave 1, palavra chave 2, etc.>

---

# Sumário

---

Lista de Figuras	5
Lista de Tabelas	6
1 Introdução	7
2 Fundamentação Teórica	8
2.1 Rede Neural Recorrente e LSTM	8
2.2 Validação cruzada K-Fold	9
2.3 Indicadores técnicos	10
2.3.1 Oscilador Estocástico (%K)	10
2.3.2 Índice de força relativa (RSI)	11
3 Proposta e Metodologia	12
3.1 Modelo da Rede Neural	12
3.2 Obtenção dos dados e features	12
3.3 Metodologia	15
3.3.1 Métricas	16
4 Testes e Resultados	18
5 Conclusões	20
Referências Bibliográficas	21

---

## Lista de Figuras

---

2.1	Uma mesma célula RNN ao longo do tempo	9
2.2	célula LSTM vista por dentro	9
2.3	Processo do K-Fold (k=5)	10
3.1	Modelo da Rede Neural	13
3.2	Fluxograma da obtenção de features	15

---

## Lista de Tabelas

---

4.1	Mean Squared Error (média)	18
4.2	Movement Accuracy (média)	18
4.3	Price Up Down Accuracy (média)	18
4.4	Média das métricas do modelo genérico	18
4.5	Desvio padrão (K-Fold)	19



## Introdução

---

O mercado de criptomoedas já capitaliza um total de 7,5 trilhões de Reais, causando aumento no interesse de investidores, que, junto do aumento no acesso à poder computacional [Hwang 2018], vêm tornando mais comum o uso de DeepLearning em aplicações de investimentos no mercado.

Apesar da previsão de mercado de ações e criptomoedas ser uma área já bem explorada, a maioria esmagadora dos trabalhos focam em treinar/ajustar seus modelos para um único mercado alvo (ação na bolsa de valores ou criptomoeda). A especialização tem seus benefícios de extrair o máximo de performance possível, porém também carregam seus pontos negativos como a incapacidade de aplicar um modelo treinado à outros alvos. Por exemplo, assim como um carro autônomo treinado para dirigir no Japão não funcionaria no Brasil sem ajustes, uma Rede Neural treinada para prever Bitcoin não teria eficácia para prever outra criptomoeda pouco conhecida.

Modelos de LSTM, comumente aplicados para problemas de NLP (*Natural Language Processing*), se mostram uma ótima escolha para previsão de criptomoedas como apontado em [Tandon et al. 2019] e [Felizardo et al. 2019].

Também foi estudada a correlação de tweets e google trends com preço do Bitcoin por [Abraham, Jethin; Higdon, Daniel; Nelson, John; e Ibarra, Juan 2018], concluindo que índices de interesse (volume de tweets e google trends) estão altamente correlacionados ao preço do Bitcoin, porém a análise de sentimentos de tweets não é um indicador confiável quando os preços estão caindo.

Sendo assim, neste trabalho vamos construir e analisar um **modelo genérico** para prever a variação do preço de criptomoedas, capaz de atingir performance comparáveis em criptomoedas nunca vistas pelo modelo antes. Também analisar o uso de features adicionais, derivadas de indicadores técnicos, dados de redes sociais e calendário de eventos dirigidos pela comunidade.

---

## Fundamentação Teórica

---

Com o aumento no poder computacional, o uso de DeepLearning vem se tornando mais acessíveis em aplicações de investimentos no mercado de ações. Modelos de RNN (*Recurrent Neural Network*), comumente aplicados para problemas de NLP (*Natural Language Processing*), também se mostraram uma ótima escolha para predição de séries temporais. Muitos modelos utilizam histórico de preços e indicadores técnicos [Weng, Ahmed e Megahed 2017] como fonte de *features*, e alguns estudos mostram a relação de dados sociais/obtidos da internet com o preço [Abraham, Jethin; Higdon, Daniel; Nelson, John; e Ibarra, Juan 2018], [Weng, Ahmed e Megahed 2017]

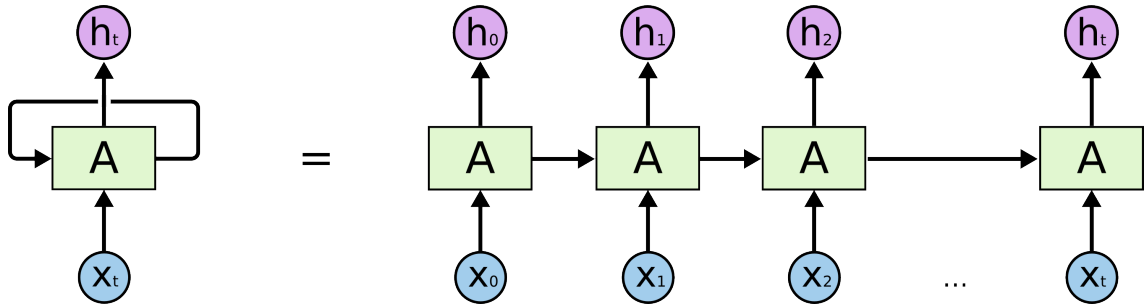
### 2.1 Rede Neural Recorrente e LSTM

Redes neurais tradicionais não possuem persistência de dados, isto é, informações não são mantidas entre etapas durante a execução. Isto afeta diretamente a natureza de séries temporais, por exemplo: dado um histórico de preços do Bitcoin, a ocorrência de uma alta á 7 dias atrás pode resultar em uma resistência próxima deste preço hoje.

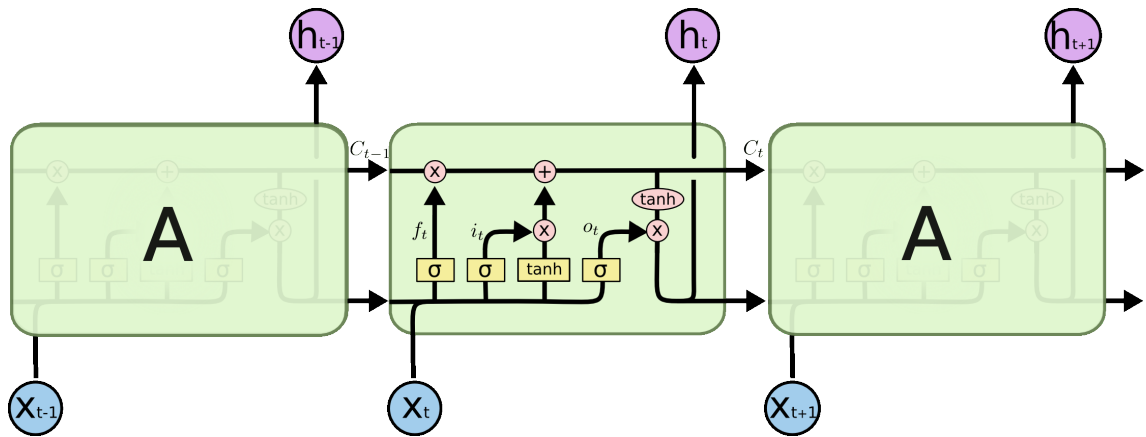
Para isso redes com recorrência (RNN) foram criadas, permitindo persistência de informação. A imagem 2.1 mostra uma célula *A* de uma RNN ao longo do tempo, onde dada uma sequência *X* como entrada tal que  $x_t$  é o elemento de índice *t* em *X*, uma célula RNN recebe a entrada  $x_t$ , gera a saída  $h_t$  e passa informação para ela mesma através do loop.

Entretanto uma RNN tem dificuldades em manter informação à longo prazo, para isso a LSTM (*Long Short Term Memory*), uma subclasse da RNN, [Hochreiter e Schmidhuber 1997] foi criada, onde por design é capaz de "lembrar" de informações por longos períodos de tempo.

Uma LSTM possui um vetor de memória  $c_t$  chamado *cell state*, e utiliza portões *input gate*  $i_t$ , *forget gate*  $f_t$  e *output gate*  $o_t$  para explicitamente controlar o valor de  $c_t$ , como mostrado na imagem 2.2 e explicado mais à fundo em [Olah 2015] e [Ming et al. 2017].



**Figura 2.1:** Uma mesma célula RNN ao longo do tempo  
imagem retirada de <https://colah.github.io>



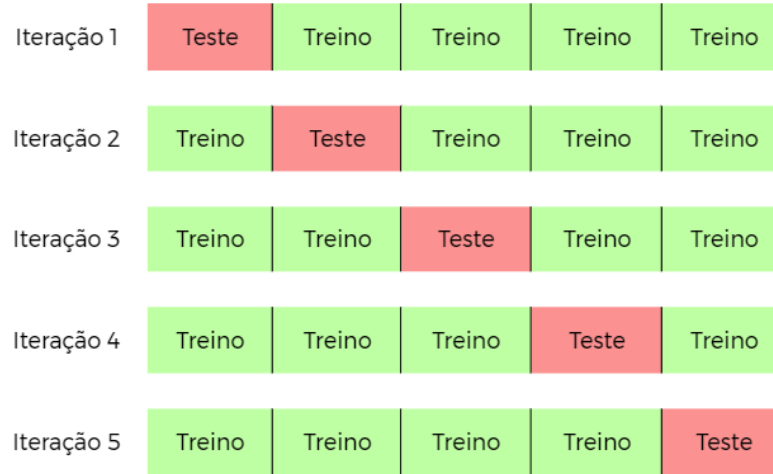
**Figura 2.2:** célula LSTM vista por dentro  
imagem retirada de <https://colah.github.io> e editada.

## 2.2 Validação cruzada K-Fold

Validação cruzada é um método estatístico usado para estimar a capacidade/performance geral de um modelo. É um método popular pela simplicidade e por resultar em estimativas de performance menos enviesadas que outros métodos.

Para isso ele testa todo o dataset através da rotação na divisão do conjunto entre treino e teste 2.3.

O método possui um único parametro  $K$ , onde  $K$  é o número de grupos que o conjunto de dados será dividido.



**Figura 2.3:** Processo do K-Fold ( $k=5$ )

Após treinar e testar todas iterações, a média da performance é calculada, e neste trabalho em adicional, também é calculado o desvio padrão para medir o quão consistente é a performance do modelo com os conjuntos de dados.

## 2.3 Indicadores técnicos

Neste trabalho vamos usar dois dos três indicadores técnicos usados em [Weng, Ahmed e Megahed 2017] (pois dentre *oscilador estocástico*, *%R de Williams* e *RSI*, os dois primeiros diferem em apenas como são escalados), devido a similaridade da natureza especulativa entre o mercado de ações e mercado de criptomoedas.

### 2.3.1 Oscilador Estocástico (%K)

$$\%K = \left( \frac{C - Low_t}{High_t - Low_t} \right) \times 100$$

- $C$  = o preço de fechamento
- $Low_t$  = o menor preço nos ultimos  $t$  dias
- $High_t$  = o maior preço nos ultimos  $t$  dias
- $\%K$  = The current value of the stochastic indicator

Definido pela fórmula acima, é um indicador de *momentum* que compara um preço específico com as máximas e mínimas recentes, retornando um valor entre 0 e 100. Ele é usado para indicar sinais de *overbought* (sobre-comprado) e *oversold* (sobre-vendido). O valor padrão de  $t$  é  $t = 14$ .

Normalmente valores acima de 80 são considerados como *overbought* e abaixo de 20 como *oversold*.

### 2.3.2 Índice de força relativa (RSI)

O RSI também é um indicador de *momentum* que retorna valores entre 0 e 100, mas diferente do Oscilador Estocástico, dá sinais de ***overbought*** e ***oversold*** medindo a velocidade do movimento de preços, ou em outras palavras, medindo a magnitude de mudanças recentes no preço.

Ele é definido pela seguinte fórmula:

$$RSI = 100 - \left( \frac{100}{1 + \frac{Average\ gain}{Average\ loss}} \right)$$

onde *Average gain* e *Average loss* são expressos em valores positivos definidos por

$$Average\ gain = \frac{((media\ dos\ ganhos\ nos\ ultimos\ t\ dias) \times (t - 1) + ganho\ atual)}{t}$$

$$Average\ loss = \frac{((media\ das\ perdas\ nos\ ultimos\ t\ dias) \times (t - 1) + perda\ atual)}{t}$$

O valor padrão de  $t$  também é  $t = 14$ .

A pratica comum entre investidores é de considerar valores acima de 70 como ***overbought*** e abaixo de 20 como ***oversold***.

Em geral o Oscilador Estocástico são mais uteis para mercados agitados que sobem e descem numa certa frequência mas sem uma tendência para uma direção, e RSI para mercados com tendência.

Desta forma, selecionando estes dois indicadores, conseguimos ter dados adequados ambos para comportamentos cíclicos e de tendência.

---

## Proposta e Metodologia

---

A proposta deste trabalho é analisar a viabilidade de um modelo genérico de Redes Neurais de Regressão para predição de criptomoedas, isto é, um modelo que após ser treinado seja aplicado para diversas outras criptomoedas sem re-treina-lo. E também analisar o uso de features extraídas de redes sociais e do calendário de eventos dirigido pela comunidade.

### 3.1 Modelo da Rede Neural

A rede é composta por 5 camadas, como mostrado em 3.1 sendo a primeira e última camadas de entrada e saída respectivamente. A primeira camada interna (2ª camada da rede) possui 750 células LSTM, a segunda camada interna 175 unidades Densas e a terceira 75 unidades Densas. Todas as camadas internas possuem *dropout* de 0.1 (10%).

Os dados de treino é uma matriz de 3 dimensões [*Amostras*  $\times$  *TamanhoJanela*  $\times$  *Features*], onde *TamanhoJanela* é a quantidade de dias anteriores que uma célula LSTM olhará. Uma Amostra (que corresponde a uma entrada na rede) possui o histórico de *TamanhoJanela* dias dos valores de cada feature.

Nossa implementação adotou *TamanhoJanela* = 30, por apresentar melhores resultados dentre os valores testados de 5, 15, 30, 60.

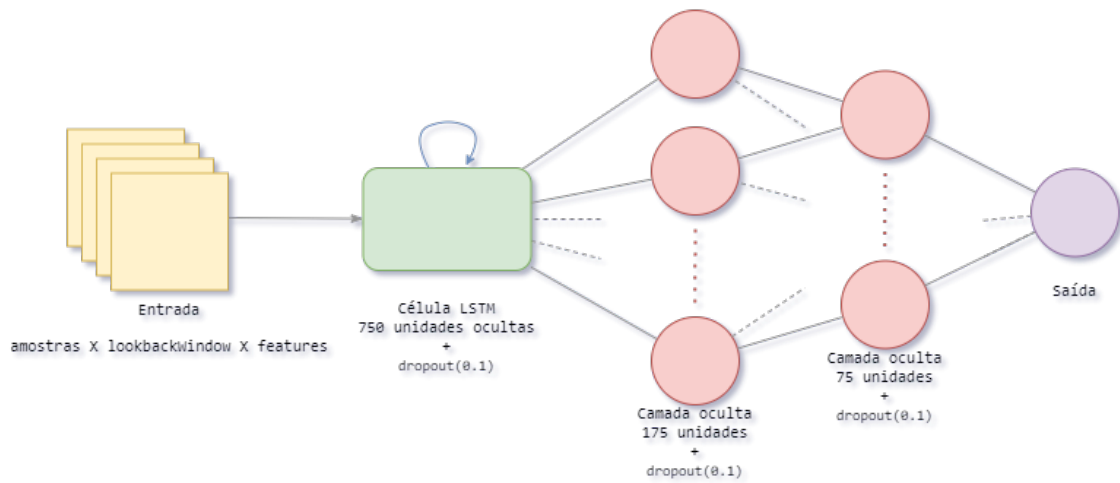
O modelo foi implementado em `python` utilizando a biblioteca Keras, e utiliza a função *loss* (função de erro) *MeanSquaredError* (Erro Quadrático Médio).

### 3.2 Obtenção dos dados e features

Foram usados dois tipos de dados:

1. **estatísticos/técnicos:** preço, Oscilador Estocástico, Índice de força relativa (RSI)
2. **dados sociais:** volume de tweets, google trends, eventos de criptomoedas;

Os dados estatísticos são obtidos de um dataset do Kaggle [Rajkumar 2021], composto por arquivos .csv de diversas criptomoedas, onde cada arquivo possui o his-



**Figura 3.1:** *Modelo da Rede Neural*

tórico de granularidade diária do preço de abertura, fechamento, alta, baixa e volume. Então é calculado o **preço**(= *abertura* − *fechamento*), e os **indicadores técnicos** usando a biblioteca `pandas-ta` no python.

Já os dados sociais são extraído de multiplas fontes 3.2:

- **Volume de tweets:** retirado dos gráficos encontrados no [bitinfocharts.com](https://bitinfocharts.com), usando um webScraper implementado com Selenium.
- **Google trends:** também retirado dos gráficos encontrados no [bitinfocharts.com](https://bitinfocharts.com), e em caso de não existir dados para a criptomoeda desejada, é utilizada a API do googleTrends com o método de reconstrução por sobreposição proposto em [TSENG 2019].
- **Calendário de eventos:** o website [coinmarketcal.com](https://coinmarketcal.com) provê calendários dirigidos pela comunidade com eventos de criptomoedas. Foi implementado um webScraper para extrair uma lista de eventos, do qual em seguida é convertido em múltiplas séries temporais:
  1. Duas séries temporais para o anúncio de um evento, uma para sua pontuação e outra para sua confiabilidade.
  2. Várias séries temporais informando que em  $n$  dias ele irá acontecer.

Dado que um evento possui os atributos [data de divulgação, data prevista para ocorrer, votos, confiabilidade], um total de  $2(N+1)$  séries temporais serão criadas, onde  $N$  = *quantos dias informar com antecedencia*.

por exemplo, dado os seguintes eventos:

	Anunciado dia	Acontece dia	Votos	Confiabilidade
Evento A	0	6	250	0.85
Evento B	5	8	300	0.99

numa série temporal de 11 dias, vamos informar a rede com **3 dias de antecedência** ( $N = 3$ ):

```
# o índice representa o dia, e o valor os votos|confiab. do evento
# dia nº          0    1    2    3    4    5    6    7    8    9   10
anuncio_de_evento=[250, 0,   0,   0,  , 0,  , 300, 0,  , 0,  , 0, 0, 0]
anunc_evento_conf=[.85, 0,   0,   0,  , 0,  , .99, 0,  , 0,  , 0, 0, 0]

evento_falta_3dia=[0,  , 0,   0,   250, 0,  , 300, 0,  , 0,  , 0, 0, 0]
evento_falta_2dia=[0,  , 0,   0,   0,  , 250, 0,  , 300, 0,  , 0, 0, 0]
evento_falta_1dia=[0,  , 0,   0,   0,  , 0,  , 250, 0,  , 300, 0, 0, 0]

evento_3dia_conf=[0,  , 0,   0,   .85, 0,  , .99, 0,  , 0,  , 0, 0, 0]
evento_2dia_conf=[0,  , 0,   0,   0,  , .85, 0,  , .99, 0,  , 0, 0, 0]
evento_1dia_conf=[0,  , 0,   0,   0,  , 0,  , .85, 0,  , .99, 0, 0, 0]
```

em caso de intersecção entre eventos na mesma série temporal, é feita a soma da pontuação e mesclagem proporcional da confiabilidade:

$$votes = v_1 + v_2$$

$$merged\ confidence = \frac{(v_1 * c_1) + (v_2 * c_2)}{v_1 + v_2}$$

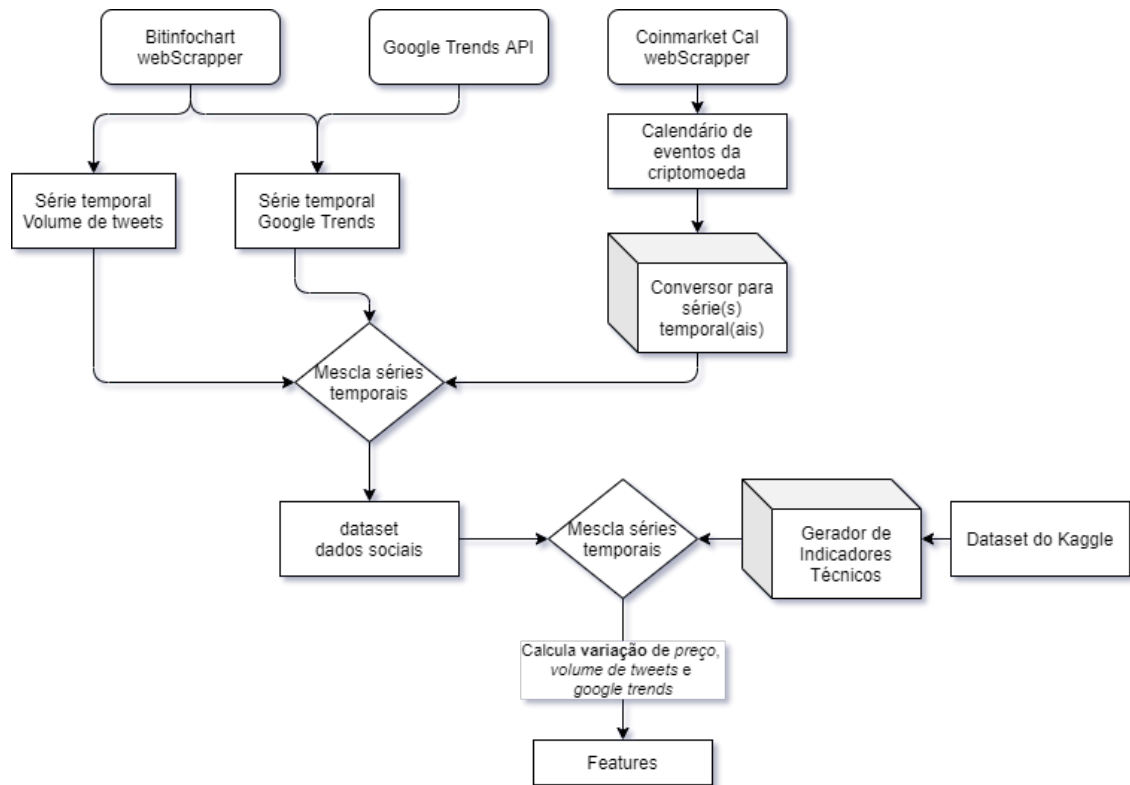
onde, dados dois eventos X e Y que se interseccionam,  $v_1$  e  $c_1$  são o *votos* e *confiabilidade* de X, e  $v_2$  e  $c_2$  são o *votos* e *confiabilidade* de Y, respectivamente.

Após extrair e mesclar todos os dados sociais num único dataset, eles são unidos com o dataset dos dados estatísticos/técnicos e calculado mais três features: *variação do volume de tweets*, *variação do google trends* e *variação do preço*, onde *variação* é a mudança em porcentagem entre o valor do *timestep* anterior e atual:

$$variacao[i] = \left( \frac{valor[i] - valor[i-1]}{valor[i-1]} \right)$$

Ao final, temos a seguinte lista de features/séries temporais disponíveis:





**Figura 3.2:** Fluxograma da obtenção de features

- |                                     |   |
|-------------------------------------|---|
| 1. Preço                            | 8. Variação do google trends (%)        |
| 2. Variação do preço (%)            | 9. Anúncio de evento - votos            |
| 3. Oscilador Estocástico            | 10. Anúncio de evento - confiabilidade  |
| 4. Índice de força relativa (RSI)   | 11. 1 dia para evento acontecer - votos |
| 5. Volume de tweets                 | 12. 1 dia para evento acontecer - conf. |
| 6. Variação do volume de tweets (%) | ...                                     |
| 7. Google trends                    | N. n dias para evento acontecer - conf. |

Com exceção das features de variação (%), todas são normalizadas no intervalo [-1, 1].

### 3.3 Metodologia

O trabalho será composto por duas partes:

1. Análise e validação do modelo genérico para previsão de criptomoedas.
2. Análise e validação do uso de features de indicadores técnicos e sociais.

Para a **análise e validação do modelo genérico**, quatro criptomoedas serão testadas: **Bitcoin**(BTC), **Ethereum**(ETH), **Tron**(TRX) e **EOS**(EOS);

As duas primeiras são moedas conhecidas de menor volatilidade, enquanto as duas segundas são *Altcoins* ("moedas alternativas") relativamente novas (lançadas no mercado

em 2017 e 2018 respectivamente), que possuem maior volatilidade quando comparadas com Bitcoin e Ethereum. O objetivo é testar como um modelo se comporta na predição de outras moedas com comportamentos de mercado similares e não-similares, isto é, outras moedas das quais não foram usadas em seu treino.

Para testar a performance base de cada criptomoeda, será usado o método de K-Fold ( $K=5$ ) com **early stopping**, treinando e testando com dados da **mesma criptomoeda**.

Então vamos testar a performance para cada combinação  $(A,B)$  tal que  $A,B \in \{BTC,ETH,TRX,EOS\}$  e  $A \neq B$ , treinando o modelo com todos os dados da criptomoeda  $A$  e testando com todos os dados da criptomoeda  $B$ , isto é, treinar e prever com **diferente criptomoedas**.

Já para a **análise e validação do uso de indicadores técnicos e de dados sociais**, será usado apenas o **Bitcoin** (BTC) como base, por ser a moeda de menor volatilidade. E então verificar o aumento e/ou perca de performance ao adicionar uma feature por vez.

### 3.3.1 Métricas

Por ser um modelo de regressão, a métrica *MeanSquaredError* (Error quadrático médio) foi adotada, mas também optamos por adicionar duas outras métricas de acurácia,

- **MeanSquaredError** (MSE): uma das métricas mais comuns
- **Movement Accuracy** (MovAcc): taxa de acerto da direção do movimento.

Para predição preço "crú" isso é equivalente à taxa de acerto de se o preço subiu ou desceu. Para predição de variação (%) do preço, é equivalente a taxa de acerto de se a % foi maior ou menor que a anterior.

$$up(prev, now) = \begin{cases} 1 & \text{se } prev < now \\ 0 & \text{se } prev \geq now \end{cases}$$

$$sameDirection(past, true, pred) = \begin{cases} 1 & \text{se } up(past, true) = up(past, pred) \\ 0 & \text{se } up(past, true) \neq up(past, pred) \end{cases}$$

$$MovAcc(A, B) = \frac{\sum_{i=1}^{|A|} sameDirection(A_{i-1}, A_i, B_i)}{|A|}$$

Onde A e B são os valores alvo e de predição do modelo, respectivamente.

- **Price Up Down Accuracy** (UpDownAcc): taxa de acerto de aumento ou diminui-

ção no preço, métrica aplicável apenas para predição de variação (%) do preço:

$$aboveZero(x) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x \leq 0 \end{cases}$$

$$sameSide(x, y) = \begin{cases} 1 & \text{se } aboveZero(x) = aboveZero(y) \\ 0 & \text{se } aboveZero(x) \neq aboveZero(y) \end{cases}$$

$$UpDownAcc(A, B) = \frac{\sum_{i=1}^{|A|} sameSide(A_i, B_i)}{|A|}$$

Onde A e B são os valores alvo e de predição do modelo, respectivamente.

## Testes e Resultados

Falta escrever aqui ainda, por enquanto é um rascunho (mas os dados nas tabelas estão corretos)

*Nota:* Não foi encontrado nenhum outro trabalho que utiliza a variação dos preços ao invés do preço bruto. E também nenhum outro que tenta generalizar os modelos

caso vc esteja confuso: A diagonal em cinza é com o K-fold (ele é usado quando treinando e testando com a mesma moeda).

as outras células da tabela não possuem k-fold, são a "pontuação" obtida para a métrica, ao treinar com todo conjunto de dados de uma **moeda base** (linhas), e testado com todo conjunto de dados de outra **moeda alvo** (colunas).

	BTC	ETH	TRX	EOS
BTC	.00039	.00086	.00088	.00090
ETH	.00040	.00086	.00089	.00090
TRX	.00039	.00086	.00088	.00090
EOS	.00040	.00086	.00088	.00090

**Tabela 4.1:** Mean Squared Error (média)

	BTC	ETH	TRX	EOS
BTC	73.42%	73.87%	73.41%	72.69%
ETH	72.37%	75.44%	73.23%	72.87%
TRX	72.05%	73.87%	75.42%	72.96%
EOS	72.25%	73.76%	73.23%	74.88%

**Tabela 4.2:** Movement Accuracy (média)

	BTC	ETH	TRX	EOS
BTC	75.55%	76.34%	76.90%	72.06%
ETH	74.58%	77.69%	76.90%	72.15%
TRX	74.34%	76.62%	78.42%	72.42%
EOS	74.26%	76.85%	76.54%	74.44%

**Tabela 4.3:** Price Up Down Accuracy (média)

**Tabela 4.4:** Média das métricas do modelo genérico

eixo y: moeda usada para treinar; eixo x: moeda testada

Desvio Padrão		BTC	ETH	TRX	EOS
	MSE	4.695 e-5	1,266 e-4	1,289 e-4	1,624 e-4
	Mov. Acc.	2,331 e-2	1,649 e-2	2,911 e-2	2,448 e-4
	UpDown Acc.	1,447 e-2	1,755 e-2	2,344 e-2	2,262 e-4

**Tabela 4.5:** *Desvio padrão (K-Fold)*

## Conclusões

---

<nota pros orientadores: essas conclusões são verdadeiras, só ainda não escrevi todos os dados no capítulo de Resultados>

Ao Utilizar a variação (%) do preço ao invés do preço em sí, conseguimos generalizar a informação do comportamento do preço, resultando nas seguintes observações:

- Preços muito pequenos e muito altos são convertidos para o mesmo "escopo" de valores, permitindo que o modelo aproveite todo o conjunto de dados e se adeque mais facilmente a todo ele, consequentemente aumentando a performance/acurácia.
- Junto da observação acima, preços de criptomoedas diferentes são trazidos para o mesmo "escopo" também, permitindo que o modelo saiba como se comportar com dados de criptomoedas nunca vistas antes (de volatilidade similar ou não-similar), tendo uma diferença de performance pequena.

Também observamos que o uso de dados sociais não melhorou a performance do modelo quando configurado para prever a variação (%) do preço. Enquanto o uso de indicadores técnicos, bem conhecidos por investidores do Mercado de Ações, provocou um pequeno aumento na performance em todas as métricas, de forma consistente.

---

## Referências Bibliográficas

---

- [Abraham, Jethin; Higdon, Daniel; Nelson, John; e Ibarra, Juan 2018]Abraham, Jethin; Higdon, Daniel; Nelson, John; e Ibarra, Juan. [Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis](#). 2018. Disponível em: <<https://scholar.smu.edu/datasciencereview/vol1/iss3/1>>.
- [Caux, Bernardini e Viterbo 2020]CAUX, M. de; BERNARDINI, F.; VITERBO, J. Short-term forecasting in bitcoin time series using lstm and gru rnns. In: . [S.l.: s.n.], 2020. p. 97–104.
- [Felizardo et al. 2019]FELIZARDO, L. et al. Comparative study of bitcoin price prediction using wavenets, recurrent neural networks and other machine learning methods. In: . [S.l.: s.n.], 2019. p. 1–6.
- [Hochreiter e Schmidhuber 1997]HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, v. 9, p. 1735–80, 12 1997.
- [Hwang 2018]HWANG, T. *Computational Power and the Social Impact of Artificial Intelligence*. 2018.
- [Jaquart, Dann e Weinhardt 2021]JAQUART, P.; DANN, D.; WEINHARDT, C. Short-term bitcoin market prediction via machine learning. *The Journal of Finance and Data Science*, v. 7, p. 45–66, 2021. ISSN 2405-9188. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2405918821000027>>.
- [Ming et al. 2017]MING, Y. et al. Understanding hidden memories of recurrent neural networks. 10 2017.
- [Nelson, Pereira e Oliveira 2017]NELSON, D.; PEREIRA, A.; OLIVEIRA, R. de. Stock market's price movement prediction with lstm neural networks. In: . [S.l.: s.n.], 2017. p. 1419–1426.
- [Olah 2015]OLAH, C. *Understanding LSTM Networks*. 2015. Disponível em: <<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>>.
- [Rajkumar 2021]RAJKUMAR, S. [Cryptocurrency Historical Prices](#). 02 2021. Disponível em: <<https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory>>.

[Tandon et al. 2019]TANDON, S. et al. Bitcoin price forecasting using lstm and 10-fold cross validation. In: . [S.l.: s.n.], 2019. p. 323–328.

[TSENG 2019]TSENG, Q. *Reconstruct Google Trends Daily Data for Extended Period*. 2019. Disponível em: <<https://towardsdatascience.com/reconstruct-google-trends-daily-data-for-extended-period-75b6ca1d3420>>.

[Weng, Ahmed e Megahed 2017]WENG, B.; AHMED, M.; MEGAHED, F. Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications*, v. 79, 02 2017.