PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS

ESCOLA POLITÉCNICA

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LUIS CARLOS MARTINS ARRUDA JÚNIOR


# PREDIÇÃO DE COMPORTAMENTO DE AÇÕES NO MERCADO FINANCEIRO USANDO REDES NEURAIS RECORRENTES LSTM

GOIÂNIA 2021

LUÍS CARLOS MARTINS ARRUDA JÚNIOR

# PREDIÇÃO DE COMPORTAMENTO DE AÇÕES NO MERCADO FINANCEIRO USANDO REDES NEURAIS RECORRENTES LSTM

Monografia de conclusão de curso apresentada ao curso de Ciência da Computação, da Escola Politécnica da Pontifícia Universidade Católica de Goiás, como requisito parcial à conclusão do curso.
Orientador(a): Prof. Max Gontijo de Oliveira

GOIÂNIA 2021

Luís Carlos Martins Arruda Júnior

**PREDIÇÃO DE COMPORTAMENTO DE AÇÕES NO MERCADO FINANCEIRO USANDO REDES NEURAIS RECORRENTES LSTM**

Este Trabalho de Conclusão de Curso julgado adequado para obtenção o título de Bacharel em Ciência da Computação, e aprovado em sua forma final pela Escola Politécnica, da Pontifícia Universidade Católica de Goiás, em ____/____/_____.

_____
Prof. Me. Ludmilla Reis Pinheiro dos Santos
Coordenador(a) de Trabalho de Conclusão de Curso

Banca examinadora:

_____
Orientador: Prof. Me. Max Gontijo de Oliveira

_____
Prof. Me. Fernando Gonçalves Abadia

_____
Profª. Ma. Lucília Ribeiro

GOIÂNIA 2021

*Para meus pais, minha eterna gratidão e carinho, por conduzirem minha vida nos caminhos da sabedoria e do conhecimento.*

## AGRADECIMENTOS

Aos meus pais, que me deram a oportunidade de cursar uma universidade.

De modo muito especial agradeço o meu orientador, Me. Max Gontijo, pela paciência, dedicação e confiança depositada, e pelas inúmeras contribuições para a minha carreira acadêmica e profissional.

Agradeço à Escola Politécnica da PUC-Goiás, pela oportunidade de realização deste trabalho.

A todos os professores, minha consideração, pois através de seus conhecimentos, auxiliaram na minha pesquisa.

Por fim, aos meus colegas pelo auxílio nas tarefas desenvolvidas durante o curso.

Agradeço a Profa. Ma. Lucília Ribeiro e o Prof. Mr. Fernando Gonçalves por aceitar compor a minha banca examinadora.

Finalmente, agradeço a todos, que direta ou indiretamente, contribuíram para a realização deste trabalho.

## RESUMO

Neste trabalho foi utilizado a Rede Neural Recorrente (RNR) *Long Short Term Memory* (LSTM) para predizer ativos financeiros do mercado de ações brasileiro. A LSTM foi aplicada em dois ativos, BBAS3 do banco do Brasil e a PETR4 da Petrobrás com quatro series temporais para cada ativo financeiro com os intervalos de tempo em 5 minutos, 15 minutos, 30 minutos e valores diários. O modelo gerado pela LSTM foi avaliado por três métricas de erro, *ROOT MEAN SQUARED ERROR – RMSE, MEAN SQUARED ERROR – MSE, MEAN ABSOLUTE ERROR – MAE*. Os resultados mostraram que a LSTM é capaz de realizar a predição de ativos financeiros. Foi desenvolvido um protótipo de uma ferramenta Web, onde uma pessoa irá receber informações de crescimento ou recuo no preço do ativo e poderá simular a compra e venda de ativos. Essa informação será fornecida pela LSTM. A ferramenta Web foi desenvolvida utilizando a linguagem Java e o Framework Angular.



**Palavras-chave:** Redes Neurais. LSTM. Mercado Financeiro. Aprendizado de Máquina.

**ABSTRACT**

In this research, the Recurrent Neural Network (RNR) Long Short-Term Memory (LSTM) was used to predict financial assets in the Brazilian stock market. The LSTM was applied to two assets, BBAS3 from Banco do Brasil and PETR4 from Petrobrás with four time series for each financial asset with time intervals in 5 minutes, 15 minutes, 30 minutes, and daily values. The model generated by LSTM was evaluated by three error metrics, ROOT MEAN SQUARED ERROR – RMSE, MEAN SQUARED ERROR – MSE, MEAN ABSOLUTE ERROR – MAE. The results showed that the LSTM can predict financial assets. A prototype of a Web tool was developed, where a person will receive information on the increase or decrease in the asset's price and will be able to simulate the purchase and sale of assets. This information will be provided by LSTM. The Web tool was developed using the Java language and the Angular Framework.

**Keywords**: Neural Networks. LSTM. Financial market. Machine Learning.

## LISTA DE FIGURAS

## LISTA DE SIGLAS

**B3** – Bolsa de Valores do Brasil

**BBAS3** – Ativo do Banco do Brasil

**CSV** – *Comma Separated Values*

**CPU** – *Central Processing Unit*

**IA** – Inteligência Artificial

**LSTM** - *LONG SHORT THERM MEMORY*

**MAE** – Mean Absolute Error

**MLP** – Multi-layer Perceptron

**MSE** – Mean-Squared-Error

**MVC** – Model, View e Controller

**PETR4** – Ativo da Petrobrás.

**RAM** - *Random Access Memory*

**RNN** – Redes Neurais Recorrentes

**RMSE** – Root-Mean-Square Error

**Sumário**

# 1. INTRODUÇÃO

Um mercado de ações eficiente, segundo Assaf Neto (2015), é caracterizado pela realidade do mercado e não por intenções individuais da oscilação de preços. E os desvios verificados entre os valores reais e de mercado devem ser aleatórios, representando igual probabilidade de um ativo ser sub ou supervalorizado em qualquer momento e que não tenha correlação com qualquer variável observável.

Alguns economistas pesquisadores que utilizam o modelo de Markowitz (2018), para otimizar carteiras no Brasil defendem a ideia de que os retornos esperados devem ser calculados por meio de dados históricos anuais dos preços das ações, devido à volatilidade dos retornos. Entretanto existem métodos que aprendem bem com uma grande quantidade de dados, e que podem ser utilizadas para predizer retornos futuros, como exemplo as redes neurais.

O problema de realizar a predição dos valores de ativos financeiros, consiste em encontrar um modelo que apresente um desempenho satisfatório ao operar no mercado acionário, sendo necessário trabalhar com métodos de análise de séries temporais e com a volatilidade para conseguir operar com um risco aceitável (NAMETALA, 2017). Assim, essa área de pesquisa possui características interdisciplinares abrangendo áreas como estatística, economia e computação. Devido a essa mistura de disciplinas o grau de dificuldade aumenta um pouco, pois será necessária uma busca mais aprofundada sobre os temas correlacionados (ASSIS, 2019).

Para abordar os problemas da predição de séries temporais, foram utilizadas técnicas de aprendizado de máquina, pois acredita-se que elas são capazes de caracterizar dados do mercado financeiro (FISCHER, KRAUS, 2017). Para a realização do trabalho foram selecionados dois ativos do mercado financeiro brasileiro, o BBAS3 (Banco do Brasil) e PETR4 (Petrobras). Estão sendo utilizados dados dos preços destes ativos de 2010 a 2020 com quatro intervalo de tempo na série temporal, sendo eles: 5 minutos, 15 minutos, 30 minutos e diário.

B3 divulgou que 2 milhões de investidores entraram na bolsa entre 2019 e 2020. De acordo com a B3, o perfil dos novos investidores é jovem, sem filhos e com renda mensal de até R$ 5 mil e trabalha em tempo integral (B3, 2021).

No trabalho busca-se explorar a capacidade de predição do preço de ativos financeiros da B3 (Brasil, Bolsa e Balcão) por meio de Redes Neurais Recorrentes *Long Short Term*

*Memory (RNR LSTM).* Onde foi realizado testes com uma variedade de dados e períodos, os testes foram avaliados utilizando as métricas de desempenho*: Root-Mean- Square Error* (RMSE), *Mean-Squared-Error* (MSE) e *Mean Absolute Error* (MAE).

O trabalho desenvolvido visa responder a seguinte questão: é possível uma rede neural ajudar os investidores do mercado de ações brasileira, com predições de ativos financeiros?

O trabalho tem como objetivo geral aplicar a rede neural LSTM em ativos financeiros e criar uma ferramenta que consiga receber os resultados da predição da LSTM e fornecer aos investidores.

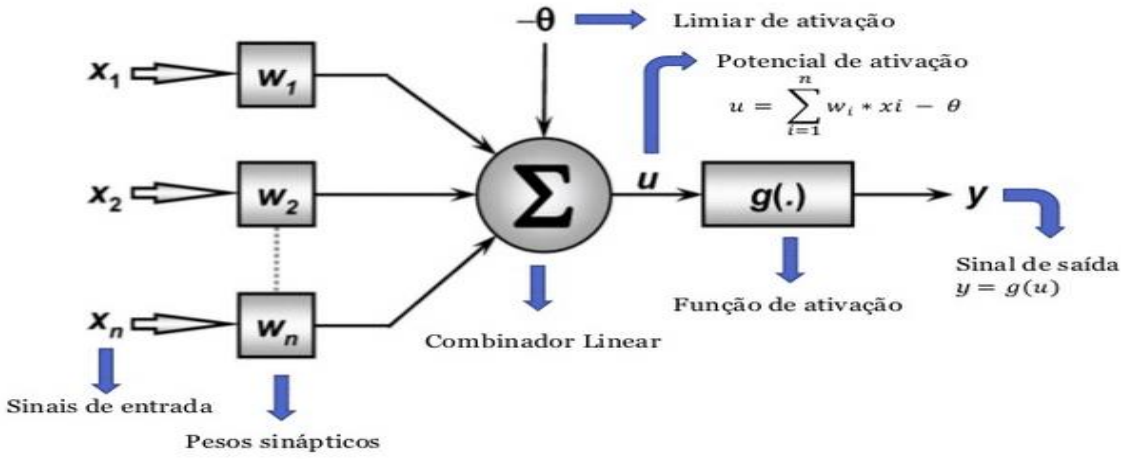
## 2. REFERENCIAL TEÓRICO

Esse capítulo será responsável por elucidar o leitor de como é o funcionamento de algumas redes neurais e o porquê da escolha da rede neural LSTM.

### 2.1 Perceptron

O *Perceptron* foi apresentado primeiramente em (Rosenblatt, 1958), sendo uma rede neural simples baseada em um neurônio biológico. O *perceptron* é constituído por um modelo matemático, onde é realizado todo o processamento dos dados e aprendizado da rede.

Figura 1– Rede *Perceptron* de uma única camada.




Fonte: PALMIERE, 2016

- O *Perceptron* (Figura 1) por ser uma rede neural clássica tem suas características bem definidas.
- A entrada é representada por $\{X_i\}$, e tem seus valores definidos como binários, ou reais, e seus valores vem de fontes externas;
- Os pesos sinápticos definidos por $\{W_i\}$ têm seus valores definidos como binários, ou reais, e seus valores são inicializados aleatoriamente;
- O limiar de ativação $\{\theta\}$ tem seus valores definidos como reais, e seu valor é inicializado aleatoriamente;
- A saída $\{y\}$ tem seu valor definido como binário, pois o *Perceptron* só consegue gerar dois resultados;

- A função de ativação {g(.)} é uma função degrau ou função degrau bipolar;

O *Perceptron* possui como técnica de aprendizado o aprendizado supervisionado, pois ele depende de um valor a ser observado como referência, para poder corrigir os seus pesos sinápticos.

Para o *Perceptron* realizar a sua aprendizagem, ele irá analisar se o resultado gerado satisfaz com a saída desejada, caso o resultado não seja satisfatório ele irá corrigir os pesos sinápticos e os limiares para que a saída convirja para o resultado esperado.

Os ajustes realizados nos pesos sinápticos podem ser obtidos conforme a formulação (1).

$$W_i^{atual} = W_i^{anterior} + \eta \cdot (\hat{y} - y) \cdot x_i^k \quad (1)$$

Onde, para fazer a atualização dos pesos sinápticos, depende-se dos pesos anteriores somados a uma constante η que define a taxa de aprendizado, multiplicado pela diferença do valor esperado ŷ pelo valor gerado ŷ, multiplicado pelos seus valores de entrada.

A implementação do algoritmo *Perceptron* é bastante simples, onde ele consiste nos seguintes passos:

1) Guardar os valores de entrada ($X_1$,...$X_n$);
2) Iniciar os pesos com valores aleatórios ($W_1$,...,$W_n$);
3) Multiplicar os valores de entrada pelos pesos;
4) Depois realizar a soma do produto e subtrair o theta (normalmente é iniciado por 1*w0 (peso randômico)
5) Esse valor gerado se chama net, irá ser aplicado em uma função degrau, onde deverá haver um limiar de ativação, para fazer a separação dos dois resultados possíveis. Ex: No caso da solução de portas lógicas, é utilizado a função degrau com o valor 0.5, então se o valor gerado pela equação net for maior ou igual a 0.5 essa função degrau irá retornar 1, caso contrário 0.

6) Deverá ser calculado um erro para verificar se o resultado esperado (y) é igual ao valor gerado (ŷ), caso o erro for 0, não realiza nenhuma mudança, caso exista erro, deverá ser implementado uma medida de diferença para realizar a alteração dos pesos.

7) Repetir do passo 3 ao 5 até o algoritmo convergir a um erro mínimo (*threshold*).

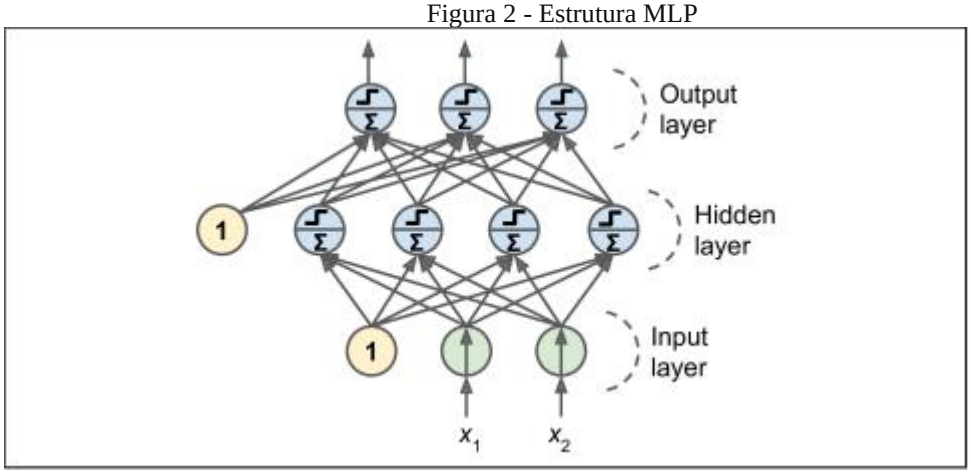
O *Perceptron* é uma rede onde ela consegue se sair muito bem quando o problema se trata em dois grupos linearmente separáveis.

Entretanto, quando se depara com um problema que não se adequa a essa característica, redes neurais artificiais mais complexas são necessárias.

## 2.2 Multilayer Perceptron (MLP)

O MLP é uma rede neural com várias camadas do *Perceptron*. E foi proposto justamente para contornar a limitação do *Perceptron* em não resolver problemas linearmente separáveis

Figura 2 - Estrutura MLP




Fonte: Géron (2019)

A Figura 2 mostra um exemplo de rede MLP. Aqui é possível verificar que a estrutura do MLP pode ser dividida em três partes: Camada de entrada (Input Layer), Camada Escondida (Hidden Layer) e Camada de saída (Output layer). A primeira parte assim como o *Perceptron* simples, é responsável por receber os dados. A segunda parte é responsável por processar a combinação dos valores de entradas com seus pesos sinápticos, a Camada escondida contém uma ou mais camadas. A última camada é responsável por gerar conjunto de saída.

O modelo de cada neurônio tem uma função de ativação não-linear. Uma forma

que pode ser utilizada para a não linearidade é a função logística:

$$y_j = \frac{1}{1+exp(-v_j)} \quad (2)$$

Onde $v$ é a soma ponderada de todas as entradas sinápticas com o limiar de ativação (bias) do neurônio $j$ e $y_j$ é a saída do neurônio.

O MLP é utilizado para resolver diversos problemas difíceis, utilizando seu treinamento supervisionado com o algoritmo *backpropagation. O backpropagation* é um algoritmo baseado na aprendizagem por correção do erro.

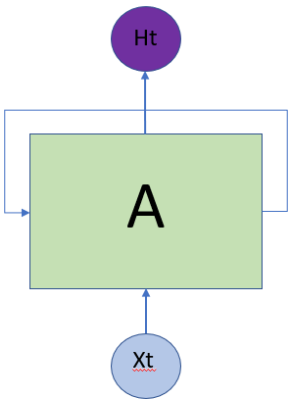
O *backpropagation* pode ser dividido em duas etapas. A propagação e a retropropagação. Na propagação, os valores de entrada percorrem toda a rede, passando camada por camada, gerando um conjunto de saídas. Já no *backpropagation*, os pesos sinápticos são adaptados de acordo com a correção de erro. Para fazer essa correção, é calculada a diferença entre o valor gerado e o valor esperado. Assim, esse erro é propagado do final para o início.

O processo de treinamento de um MLP segue a ideia de submeter diversas amostras independentes a rede e realizar a correção dos erros até que a rede seja capaz de calcular corretamente todas (ou um número suficientemente aceitável) as saídas para cada entrada. Percebe-se que, nesse contexto, as amostras não são analisadas de maneira que possam ter uma relação de causalidade uma para com a outra. Portanto, mesmo o MLP não é suficiente para problemas onde existe alguma relação de dependência entre as amostras, como uma ordem em que ocorrem no tempo. Para problemas assim seria, portanto, necessário a utilização de redes neurais recorrentes.

## 2.3 Redes Neurais Recorrentes

As redes neurais recorrentes (RNN), diferentemente das redes neurais simples, possuem a capacidade de armazenar valores ao longo do tempo. Então elas não descartam tudo antes de começar uma nova etapa.

Figura 3: Arquitetura RNN




Fonte: Autoria Própria (2021)

A RNN Figura 3 consegue trabalhar com o $X_t$ que é a entrada atual, manipulada pelo passado recente $H_{t-1}$ e assim gerar o novo $H_t$. As RNNs de maneira geral, conseguem armazenar valores do passado através desta execução cíclica de seus dados.

Esse tipo de arquitetura que torna possível trabalhar com valores do passado, faz com que as RNNs sejam comumente usadas para problemas temporais, reconhecimento de fala, processamento de linguagem natural, tradução de linguagem.

Diferentemente do *Perceptron* Simples as RNNs têm como utilização mais comum de função de ativação, a Sigmóide, Tanh e Relu, pois elas conseguem tratar problemas de não linearidade.

A arquitetura da RNN pode sofrer variações assim, surgindo vários modelos de RNN, como a Redes Neurais Recorrentes Bidirecionais (BRNN). Memória Longa a Curto Prazo (LSTM), Unidades recorrentes com portas (GRUs), dentre outros modelos arquiteturais de redes neurais recorrentes.

Esse armazenamento de valores do passado, para um RNN simples é bem simplório, então a RNN simples consegue armazenar valores do passado, mas de curto prazo. E para resolver esse problema existe a *RNN Long Short Term Memory*, onde ela consegue trabalhar com valores de um passado mais antigo.

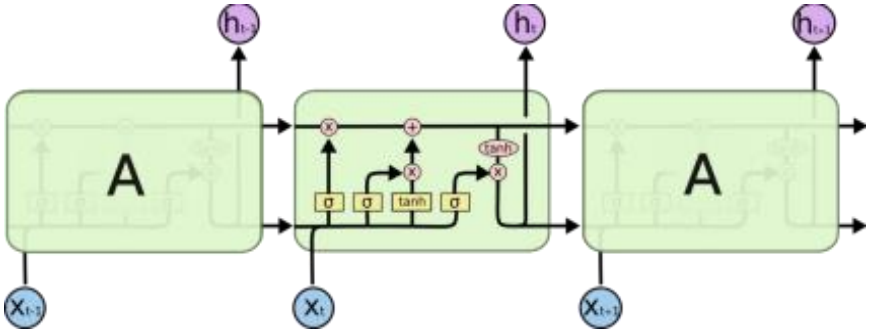
## 2.4 Long Short-Term Memory (LSTM)

A LSTM foi proposta em 1997 por *Sepp Hochrereiter e Jurgen Schmidhuber*. A

principal ideia da rede LSTM é que ela pode aprender o que armazenar no estado de longo prazo, o que jogar fora e o que ler a partir dele (Géron, 2019). Devido a esse padrão de aprendizado, ela possui excelentes resultados em capturar padrões de longo prazo em séries temporais.

Primeiro será definida a arquitetura de uma célula da rede LSTM.

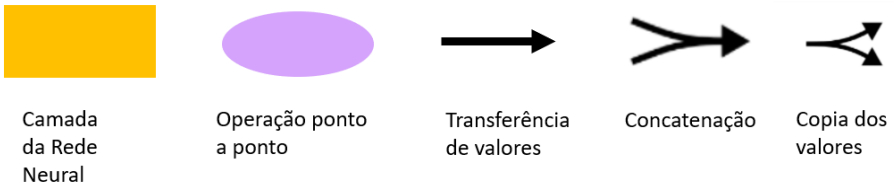
Figura 4: Arquitetura LSTM




Fonte: F Junior (2021)

A Figura 4 mostra uma arquitetura LSTM, onde é possível verificar três células LSTM concatenadas.

Figura 5: Legenda de uma célula LSTM



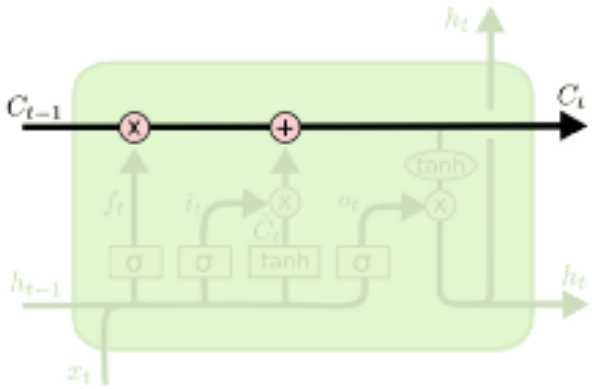

Fonte: Autoria Própria.

A LSTM tem a seguinte legenda, assim como mostra a figura 5. A camada da Rede Neural é representada por um quadrado amarelo. A operação ponto a ponto é representado pelo círculo roxo. A transferência de valores é representada pela seta para a direita. A concatenação é representada pela junção de duas retas. A cópia de valores é representada por uma bifurcação.

Uma célula LSTM é dividida em algumas etapas, as etapas serão mostradas a seguir:

- Estado da célula:

Figura 6: Estado da célula




Fonte: F Junior (2021)

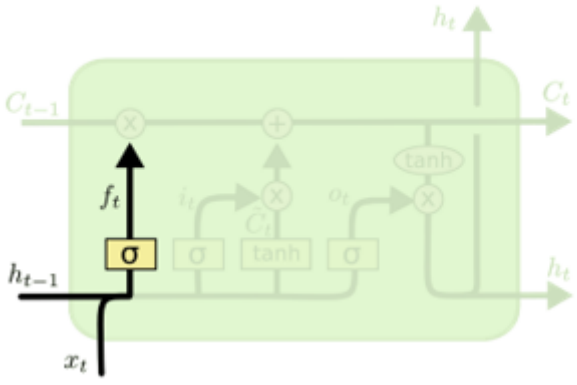
O estado da célula representado na Figura 6 é uma das principais partes da LSTM, visto que ela irá armazenar todos os valores relevantes durante toda a execução da LSTM.

- *Gates*

A LSTM consegue inserir e remover valores do estado da célula, e ela realiza essas operações através dos Gates.

Dentre esses *gates* estão: *Forget Gate, Input Gate e Output Gate*. Cada um deles possuem uma função específica na célula LSTM e de acordo com Phi (2018) eles possuem o seguinte comportamento.

Figura 7: *Forget Gate*



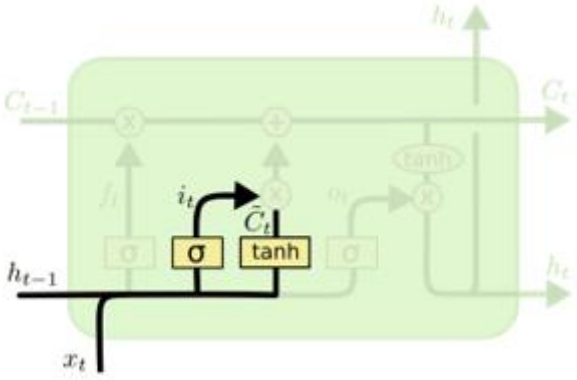

Fonte: F Junior (2021)

*Forget Gate:* o *Forget Gate* representado na Figura 7 tem como objetivo controlar quais informações devem ser retiradas no estado de longo prazo.

Esse *Gate* faz essa operação através de uma função de sigmoide. Onde ao passar valores por essa função, o resultado irá ser no intervalo de 0 e 1. Onde mais próximo de 1 é mais relevante e mais próximo de zero menos relevante.

Para o trabalho foi de grande relevância a utilização deste *Gate*, tendo em vista que foram utilizados uma grande quantidade de valores. Logo, esse portal foi de grande relevância para treinar o modelo.

Figura 8: *Input Gate*

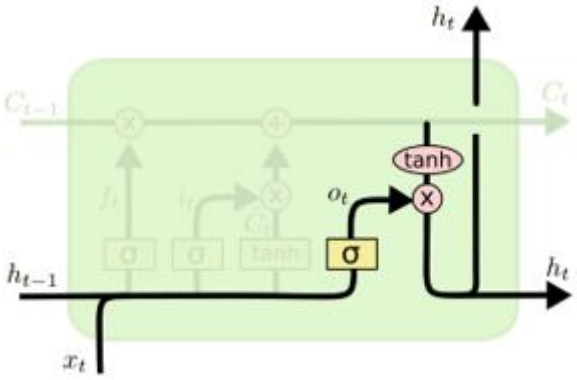



Fonte: F Junior (2021)

*Input Gate*: o *Input Gate* representado na Figura 8 tem como objetivo controlar quais informações devem ser adicionadas ao estado de longo prazo.

Para o *input Gate* assim como no *forget Gate* é utilizado uma função de sigmoide onde avaliar o que é de relevância para ser inserido no estado da célula, mas nesse caso é adicionado uma nova operação. Essa operação é nomeada como *cell candidate* onde é utilizado uma função de tangente hiperbólica (tanh) para realizar a avaliação de quais valores devem ser inseridos.

Figura 9:*Output Gate*



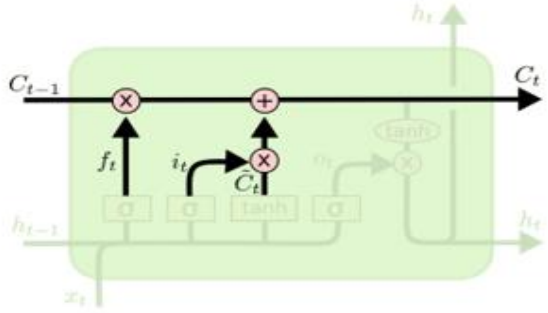

Fonte: F Junior (2021)

*Output Gate*: o *Output Gate* representado na figura 9 tem como objetivo controlar qual será o próximo estado oculto.

Agora para gerar o valor de saída é utilizado a tanh com o valor atual da *cell candidate* onde serão analisados os valores que serão produzidos.

Existem outras etapas na célula LSTM como a inserção dos *Gates* no estado da célula:

Figura 10: Inserção de valores no estado da célula




Fonte: F Junior (2021)

*Candidate Value:* o *Candidate Value* representado na Figura 10 tem como função escolher valores que serão adicionados ao *Cell State* junto com o *Input Gate.*

Os *Gates* têm um papel fundamental de o que deve deixar passar para o estado da célula, eles fazem esse controle através de funções sigmoides. Onde será gerado valores entre 0 e 1, onde zero significa baixa relevância e um significa alta relevância. Logo, zero retira, um deixa passar.

## 2.5 Métricas de Desempenho

Para verificar o desempenho de Redes Neurais em problemas de séries temporais é comum utilizar métricas que comparem os valores preditos com valores reais. Após uma pesquisa na literatura foi possível entender que existem algumas métricas que são bastantes utilizadas, então para acompanhar a literatura foram utilizadas as métricas:

### 2.5.1Mean Squared Error – MSE

- *O Mean Squared Error* calcula a diferença entre a série predita e a real ao quadrado e tira a média de seus valores. Sua fórmula é apresentada na equação 3.

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(x_t - \tilde{x}_t)^2 \quad (3)$$

### 2.5.2 Root Mean Squared Error – RMSE

- O *Root Mean Squared Error* indica a raiz do erro médio quadrático entre a série predita e a real. Sua fórmula é apresentada na equação 4.

$$RMSE = \sqrt{\sum_{t=1}^{N}(x_t - \tilde{x}_t)^2} \quad (4)$$

### 2.5.3 Mean Absolute Error – MAE

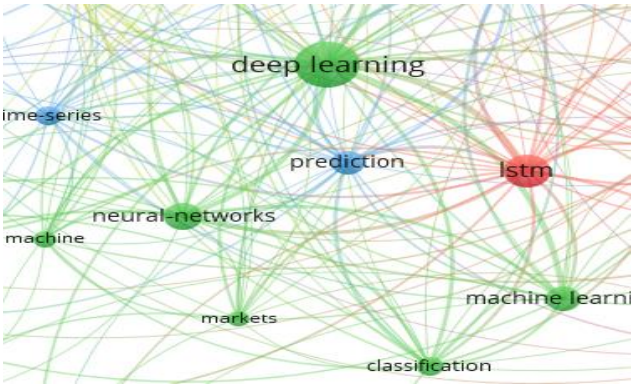
- O *Mean Absolute Error* indica o erro médio entre a série predita e a real. Sua fórmula é apresentada na equação 5.

$$MAE = \frac{\sum_{t=1}^{N}|x_t - \tilde{x}_t|}{n} \quad (5)$$

## 2.6 Trabalhos relacionados

Trabalhos com redes neurais estão bastante presentes na literatura. Foi realizada uma pesquisa na base de pesquisa *Web of Science* com a seguinte *string* de busca: Deep Learning AND LSTM AND MARKET. Foi aplicado um filtro no resultado para buscar apenas artigos publicados nos últimos 5 anos.

Figura 11: Exibição dos trabalhos encontrados por tópicos.




Fonte: Autoria Própria (2021)

Para ter uma análise dos trabalhos gerados, foi utilizado o *software livre VOSVIEWER* onde é possível inserir as referências geradas na base de pesquisa. Após inserir as referências é gerado uma nuvem de vértices (temas) conectados por arestas (relacionamento com o tema). Assim é possível ter uma ideia de quais são os trabalhos que estão mais relacionados entre si, e quais foram os temas mais abordados para aquela *string* de busca.

Com essa busca, foi possível encontrar 116 artigos. Após a seleção dos artigos foram realizados alguns passos para então filtrar quais os trabalhos foram utilizados como referência.

Para filtrar os artigos, foi primeiro realizado uma verificação de aderência nos títulos do trabalho, ou seja, verificar se o título do trabalho tem alguma conexão com o tema da pesquisa. Após essa seleção foi feita uma outra filtragem, onde foi analisado o resumo dos trabalhos, confirmando assim, a aderência dos resumos com o tema da pesquisa. Por último foi realizada uma verificação da disponibilidade dos artigos, onde só foram selecionados os artigos que estariam acessíveis livremente, ou seja, de maneira gratuita.

Como resultado, 8 artigos foram selecionados para compor a base deste trabalho, tendo como principal referência o trabalho de *Fisher* e *Krauss* que já foi citado em mais de 180 outros trabalhos.

*Fisher* e do *Krauss* (2018*)* é apresentada uma abordagem com a utilização de redes neurais recorrentes LSTM aplicada ao mercado financeiro. Foi utilizada uma série de dados de 1993 a 2015 e, como resultado, a série foi separada em três tempos. Primeiro 1993~2000, onde a LSTM teve resultado superior ao algoritmo de *Random Forest*. Segundo

2001~2009, onde a LSTM ainda foi capaz de gerar retornos positivos, mas bem inferior à década de 90. Por fim. 2010~2015, onde a LSTM continua a gerar resultados com alta acurácia.

*Fisher e Krauss* chegaram à conclusão de que a partir de 2010, os mercados apresentaram um aumento de eficiência em relação aos métodos de aprendizado de máquina, diminuindo o lucro.

Um outro trabalho de grande importância é o trabalho (BAO; YUE; RAO, 2017). A LSTM foi aplicada em alguns índices CSI 300 no mercado de ações A da China continental, *Nifty 50* no mercado de ações da índia, *Hang Seng* no mercado de Hong Kong, *Nikkei 225* em Tóquio, *S & P 500 e DJIA* na bolsa de valores de Nova York.

Os resultados foram bastante animadores, visto que essa combinação de métodos gerou resultados positivos. A receita média anual do modelo na China continental chegou a 63% e 45% no mercado da Índia (BAO; YUE; RAO, 2017)

# 3. MERCANDO FINANCEIRO

Neste capítulo será abordado brevemente sobre conceitos de mercado de ações.

## 3.1 Bolsa de Valores

A bolsa de valores é o mercado onde são realizadas as negociações de diferentes produtos de investimento; Além das transações de ações na bolsa de valores, também é realizado investimentos em fundos imobiliários, contrato com dólar, ouro, soja, entre outros.

## 3.2 Ação

Ação é a menor fração de uma empresa de capital aberto, que é negociada na bolsa de valores. Logo, ao realizar a compra uma ação, está comprando uma parte da empresa. Essas podem ser divididas em duas categorias de acordo com as suas características, direitos e riscos que ela gera aos acionistas. Sendo elas:

Ações ordinárias (ON): As ações ordinárias são o tipo de ação que dá direito ao voto na empresa.

Ações preferenciais (PN): Ações preferenciais são o tipo de ação que dá prioridade no pagamento de dividendo, e recebimento em caso de falência.

## 3.3 Brasil, Bolsa e Balcão (B3)

A B3 foi criada em 23 de agosto de 1890 por Emílio Rangel Pestana, originalmente ela foi criada com o nome de Bolsa Livre. A ideia de Emilio era a prestação de serviços inéditos, como compra e venda de títulos (B3:...2021).

Por questões políticas a bolsa teve que ser fechada e só voltou em 1895 como Bolsa de Fundos Públicos de São Paulo. Após passar por uma reformulação em 1935 a bolsa mudou de nome novamente, agora para a Bolsa de Valores de São Paulo.

A B3, como uma bolsa de valores, tem o papel de organizar o mercado de ações. Ela tem o poder de escolher as empresas que irão participar das negociações. Ela pode ser relacionada a uma grande loja, onde organiza quais são os produtos que podem ser negociáveis, e dá a oportunidade do investidor se aventurar pelo mercado financeiro.

# 4. MATERIAIS E MÉTODOS

Essa seção irá apresentar todos os materiais e métodos necessários para o trabalho.

## 4.1 Tecnologias

Para o desenvolvimento do trabalho foram utilizadas as seguintes tecnologias:

### 4.1.2Java

Java é uma linguagem de programação que foi criada na década de 90, onde ela é sustentada no paradigma de orientação a objeto (POO). O Java foi utilizado no trabalho para o desenvolvimento da ferramenta Web.

### 4.1.3 Python

Python é uma linguagem de programação que é amplamente utilizada em técnicas de machine learning, inteligência artificial, análises de dados e outras áreas. O Python foi utilizado para a utilização da rede neural LSTM.

### 4.1.4 Keras

É uma biblioteca que tem suporte para o python. Essa biblioteca pode ser utilizada para poupar implementações de redes neurais, tendo em visto que a blibioteca Keras tem um grande suporte para diversas redes neurais.

### 4.1.5 Numpy

*É* uma biblioteca do python onde ela é utilizada para fazer operações matemáticas, e devido a vasta quantidade de conteúdo relacionado a análises de dados, essa biblioteca foi escolhida para fazer o tratamento dos dados.

### 4.1.6 IDE

Pycharm é uma IDE desenvolvida pela JetBrains, e ela foi utilizada como plataforma de programação das redes neurais. Intelijj é uma IDE desenvolvida pela JetBrains, e ela foi utilizada como plataforma de programação para a ferramenta web.

### 4.1.7 Banco de Dados

Visando armazenar os valores gerados, foi selecionado o banco de dados postgresql, onde ele ficou responsável por armazenar os dados gerados pela LSTM.

### 4.1.8 Angular

Para a produção do ambiente web foi utilizado o framework frontend Angular na versão 11. Angular é um framework que foi desenvolvida pelo Google, onde esta tecnologia é responsável por gerar componentes para a elaboração do frontend.

### 4.1.9 Computador

Os testes foram realizados em um computador com processador AMD *Ryzen* 5 3400G, CPU 3.7GHz com 4 núcleos e 8 *threads*, com 16 GB de memória RAM DDR4 2666Mhz, placa de vídeo dedicada GTX 1650 OC *Windforce* 4GB e utilizando como sistema operacional o *Windows* 10.

## 4.2 Modelagem

Foi realizada uma análise dos ativos financeiros BBAS3 (BANCO DO BRASIL) e PETR4 (PETROBRAS), onde foram utilizados dados do valor de fechamento dos ativos no período de 2010 a 2020. Esses dados foram obtidos através da plataforma paga Thompson.

A arquitetura da LSTM possui: 1 neurônio para a camada de entrada, 4 blocos LSTM e uma camada de saída que gera um valor único. A rede é treinada com 100 épocas, e possui 70% dos dados para treinamento e 30% dos dados para testes.

Antes de inserir os dados na LSTM para então fazer o treinamento do modelo, é necessário definir um intervalo de aprendizado, que é basicamente a definição do tamanho dos vetores que serão inseridos na LSTM. Onde a base de dados com intervalo de tempo em 5 minutos foi utilizada com um intervalo de aprendizado com 2016 valores anteriores que dá o total de 7 dias. As bases de dados com os intervalos de tempos em 15 minutos e 30 minutos utilizam o intervalo de aprendizado com 50 valores anteriores. Para a base de dados com intervalo de tempo em 1 dia ele possui o intervalo de aprendizado com apenas 1 valor anterior. Os valores foram definidos de maneira arbitrária sem seguir nenhum padrão encontrado em outros trabalhos.

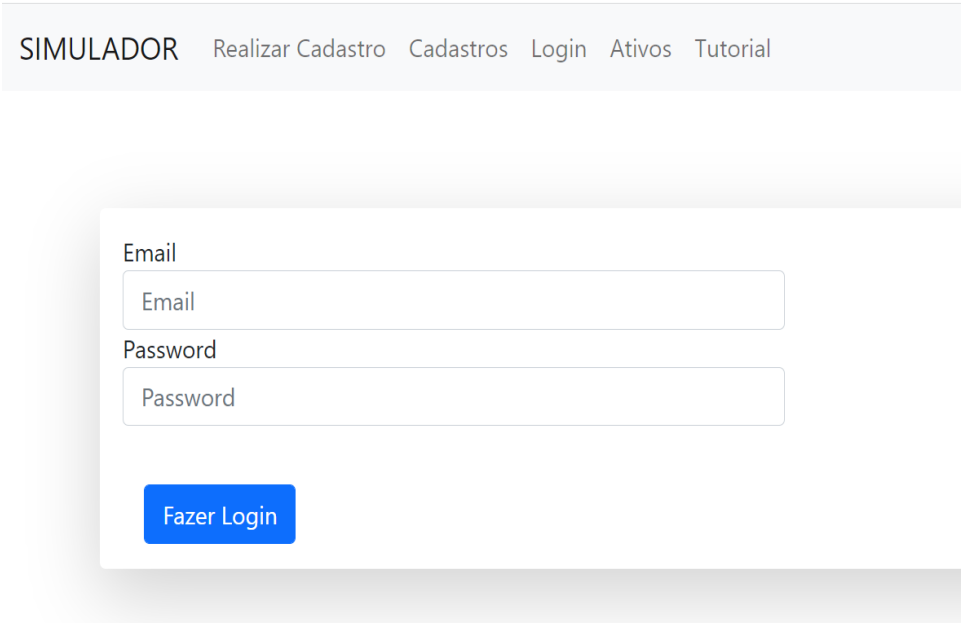
Os *datasets* possuem apenas duas colunas, uma para dia e hora e outra coluna para o valor de fechamento.

## 5. FERRAMENTA WEB

Durante o trabalho foi desenvolvido uma ferramenta web, onde é possível realizar compra e vendas de ações na série temporal que foi predita. Logo, é possível mostrar como seria o ganho ou a perda se fosse utilizado dos valores preditos pela LSTM para realizar operações na bolsa.

A ferramenta está operando em ambiente WEB, onde ele foi desenvolvido com o *framework* Angular para o *frontend* e para o *backend* desenvolvido na linguagem Java. Foi utilizado o padrão de projeto MVC (*Model, View e Controller*) com algumas alterações.

Figura 12: Tela de cadastro.

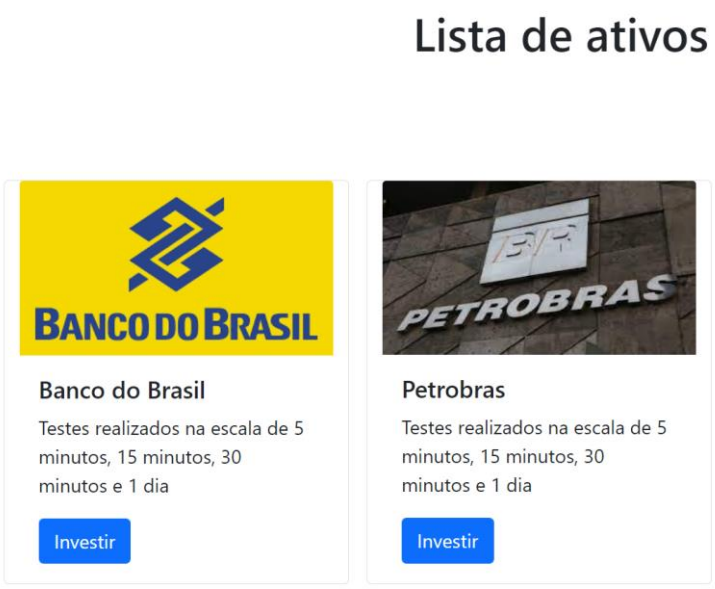



Fonte: Autoria Própria (2021)

Para realizar a utilização da ferramenta é necessário efetuar um cadastro conforme a Figura 12, e posteriormente o usuário irá escolher um ativo na lista de ativos (Figura 12) no qual ele quer fazer investimentos, e após a escolha do ativo é possível realizar compra e vendas de ações. A aplicação está operando de maneira manual, ou seja, diferente do que

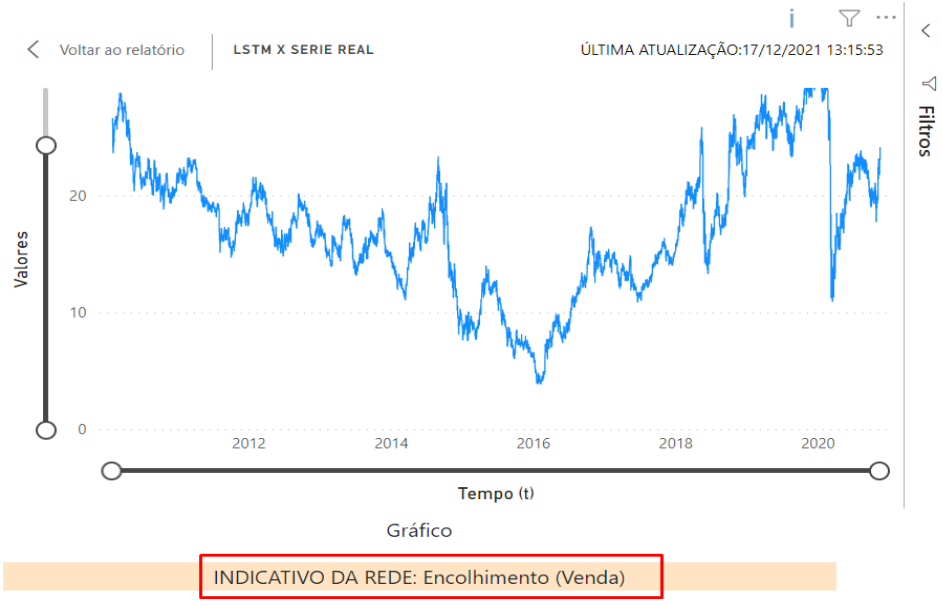
foi utilizado para realizar testes na pesquisa, pois para realizar um teste que não houvesse interação humana, foi necessário criar um mecanismo de operação manual. Hoje a aplicação está apenas indicando ao usuário, o que a rede neural prediz. Logo, toda a operação de compra e venda é realizada pelo usuário.

Figura 13: Lista de ativos




Fonte: Autoria Própria (2021)

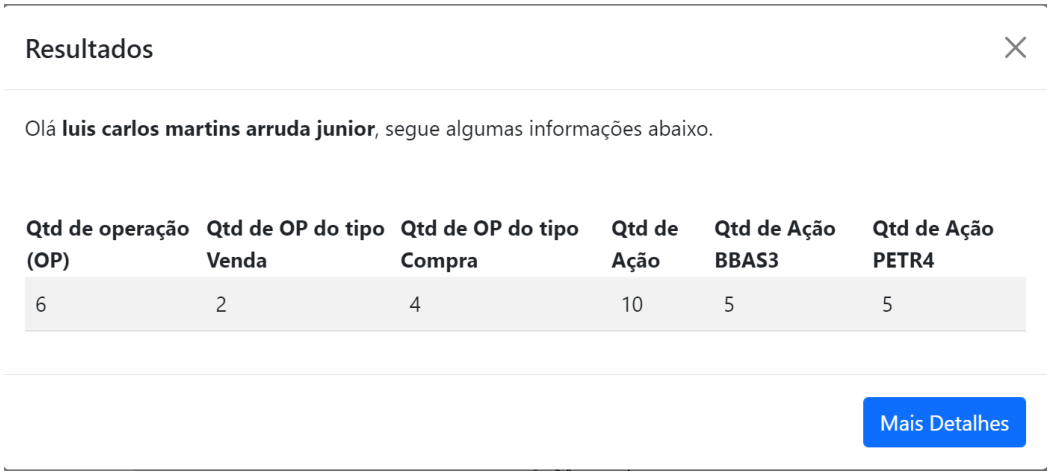
Figura 14: Indicativo da Rede




Fonte: Autoria Própria (2021)

A Figura representa os ativos que estão disponíveis para a ferramenta. Sendo assim, a aplicação será apenas um indicador de sinais (Figura 14), onde ele consegue indicar ao usuário quando ele deve comprar ou vender as ações em um momento futuro.

Figura: 15 Resultados




Fonte: Autoria Própria (2021)

Figura 16: Controle de ações


Fonte: Autoria Própria (2021)

Ao final da ferramenta é possível verificar algumas informações (Figura 15 e Figura 16), como: quantas ações possui hoje; quantas ações de cada ativo possui; qual o valor irá ter se vender todas as ações; quantas operações já realizou; e quantas de cada tipo (Compra ou venda). Além disso, a aplicação fornece a porcentagem de erro pela aplicação na fase de treinamento e de teste daquela série temporal.

## 6. RESULTADOS

Após a etapa de modelagem, a LSTM gera os resultados da predição. Para então validar os resultados gerados, foram feitas a análises dos resultados através de três métricas de erro (RMSE, MSE e MAE).

Tabela 1: Erros de cada modelo gerado pela LSTM,

| | RMSE/TRAIN | RMSE/TEST | MSE/TRAIN | MSE/TEST | MAE/TRAIN | MAE/TEST |
|---|---|---|---|---|---|---|
| PETR4_M5 | 0,05 | 0,07 | 0,00 | 0,01 | 0,03 | 0,04 |
| PETR4_M15 | 0,10 | 0,14 | 0,01 | 0,02 | 0,07 | 0,10 |
| PETR4_M30 | 0,11 | 0,18 | 0,01 | 0,03 | 0,08 | 0,11 |
| PETR4_DAILY | 0,39 | 0,66 | 0,15 | 0,44 | 0,29 | 0,46 |
| | | | | | | |
| BBAS3_M5 | 0,06 | 1,58 | 0,00 | 2,51 | 0,04 | 1,00 |
| BBAS3_M15 | 0,09 | 1,92 | 0,01 | 3,69 | 0,06 | 1,27 |
| BBAS3_M30 | 0,13 | 0,73 | 0,02 | 0,53 | 0,08 | 0,52 |
| BBAS3_DAILY | 0,55 | 2,70 | 0,31 | 7,28 | 0,41 | 2,11 |

Fonte: Autoria Própria.

Para a produção da Tabela 1, foram utilizados valores de erros obtidos nos treinamentos e testes dos modelos para cada *dataset*.

Na Tabela 1 mostra-se os erros obtidos, onde M5 caracteriza os intervalos de tempo de cinco minutos, M15 15 minutos, M30 30 minutos e *DAILY* valor diário. Para cada instância de tempo existem seis colunas de erros. Foram utilizadas três métricas de erro, RMSE, MSE e MAE. Também na Tabela 1 são apresentados os erros obtidos no treinamento e na fase de testes. Levando em consideração que quanto mais perto de 0.0 melhor o comportamento da predição.

Para validar o desempenho da LSTM foi desenvolvido uma ferramenta em Python. Essa ferramenta é responsável em realizar operações de compra e venda na série temporal predita.

A ferramenta web foi adaptada para uma espécie de um jogo, onde o jogador começa com uma quantia e ele pode ir comprando e vendendo as ações que ele tem, e no final do jogo é informado quanto de lucro ele teve. E para o jogador não ir chutando se deve comprar ou vender, a ferramenta informa o “Jogador” qual seria o possível preço da ação em um determinado tempo a frente, indicando se ele deveria comprar ou vender aquele ativo.

O tempo do jogo é dividido em momentos, que são basicamente os intervalos de tempo da série temporal. Ex: o momento 1, é o a primeira instância de tempo da série, e o momento 2 seria a segunda instancia de tempo. Sendo assim é possível se locomover na série temporal de acordo com o intervalo da série temporal, se foi utilizado uma série temporal com intervalo de cinco minutos, então o momento será separado por cinco minutos.

Para que a ferramenta possa informar ao "jogador" se o preço irá subir ou descer, foi utilizado uma técnica simples de programação, que é a manipulação de vetores, onde conseguimos acessar uma posição específica do vetor e saber qual é o valor do conteúdo naquela posição. Sendo assim é possível observar o futuro daquele vetor, e indicar ao "jogador" como é o preço daquele ativo em relação ao seu preço atual.

A ferramenta além de informar ao "jogador" como seria o possível valor do ativo em momentos futuros, também informa como seria a margem de lucro dele, pois como foi utilizado o valor da série predita, é possível fazer um comparativo entre o valor real, e o valor estimado pela LSTM.

A Tabela 2 e 3, mostra o desempenho da ferramenta em ajudar os investidores a ter lucro. Para validar o ganho dos investidores foram realizadas simulações com momentos aleatórios, onde a ferramenta possui as seguintes instruções:

- A cada execução, a ferramenta gera 3 momentos de tempo aleatórios, esses momentos serão utilizados para realizar as operações indicadas pela ferramenta.
- É criado o momento futuro, que é definido como 1000 momentos (tempo) depois que o atual.
- Se o preço atual for menor que o preço no momento futuro, ele realiza a compra de um único papel, e realiza a venda do mesmo papel quando chegar no momento predito.
- Se o preço atual for maior que o preço no momento futuro, ele não faz nenhuma entrada.

Visto as operações realizadas de maneira randômica é possível mostrar a ajuda que a LSTM irá levar aos investidores.

Tabela 2: Simulação de operação PETR4 com intervalo de tempo de 5 minutos.

| Execução | Momento (tempo) | Indicação | ValorReal | ValorPrevisto | Lucro Real | Lucro na ferramenta |
|---|---|---|---|---|---|---|
| 1 | 58881 | Compra | 21,69 | 21,72 | 0,08% | 0,08% |
| | 59881 | Venda | 22,29 | 22,29 | | |
| | 61911 | Não entrar | 23,47 | 23,35 | | |
| | 62911 | Não entrar | 22,18 | 22,14 | | |
| | 65175 | Compra | 19,88 | 19,93 | | |
| | 66175 | Venda | 20,06 | 20,13 | | |
| 2 | 22050 | Não entrar | 24,99 | 24,94 | 0,22% | 0,22% |
| | 23050 | Não entrar | 25,00 | 24,77 | | |
| | 43932 | Compra | 25,85 | 25,86 | | |
| | 44932 | Venda | 28,07 | 28,10 | | |
| | 64118 | Não entrar | 21,66 | 21,65 | | |
| | 64118 | Não entrar | 19,73 | 19,78 | | |
| 3 | 11032 | Compra | 20,43 | 20,44 | 0,17% | 0,17% |
| | 12032 | Venda | 20,65 | 20,67 | | |
| | 41362 | Compra | 24,97 | 24,95 | | |
| | 42362 | Venda | 26,41 | 26,39 | | |
| | 64510 | Não entrar | 20,14 | 20,15 | | |
| | 65510 | Não entrar | 20,07 | 20,07 | | |

Fonte: Autoria Própria (2021)

Na Tabela 2 é possível verificar a semelhança do lucro gerado entre os valores reais e o valor predito, sendo assim é possível afirmar que a rede LSTM foi capaz de aprender o comportamento da série temporal utilizada no treinamento PETR4 com intervalo de 5 minutos, e com o uso da ferramenta é possível melhorar o ganho dos investidores de ativos financeiros.

Para as três execuções totalizando 9 operações, a ferramenta conseguiu obter um lucro médio em 0.15% tanto para os dados reais quanto para a predição da LSTM, assim mostrando novamente a precisão da LSTM para o ativo da PETR4.

Tabela 3: Simulação de operação BBAS3 com intervalo de tempo de 5 minutos.

| Execução | Momento (tempo) | Indicação | ValorReal | ValorPrevisto | Lucro Real | Lucro na ferramenta |
|---|---|---|---|---|---|---|
| 1 | 1374 | Compra | 14,46 | 14,38 | 0,23% | 0,22% |
| | 2374 | Venda | 15,19 | 15,22 | | |
| | 38698 | Não entrar | 26,86 | 25,58 | | |
| | 39698 | Não entrar | 26,82 | 25,55 | | |
| | 65916 | Compra | 20,01 | 20,03 | | |
| | 66916 | Venda | 20,62 | 21,49 | | |
| 2 | 1226 | Compra | 14,37 | 14,33 | 0,17% | 0,15% |
| | 2226 | Venda | 15,11 | 15,16 | | |
| | 37139 | Compra | 26,00 | 24,96 | | |
| | 38139 | Venda | 27,01 | 25,69 | | |
| | 43180 | Não entrar | 26,88 | 25,56 | | |
| | 44180 | Não entrar | 25,86 | 24,92 | | |
| 3 | 12388 | Compra | 19,80 | 19,82 | 0,23% | 0,20% |
| | 13388 | Venda | 20,83 | 20,78 | | |
| | 25114 | Compra | 22,89 | 22,65 | | |
| | 26114 | Venda | 23,93 | 23,47 | | |
| | 40885 | Compra | 24,69 | 24,08 | | |
| | 41885 | Venda | 24,92 | 24,26 | | |

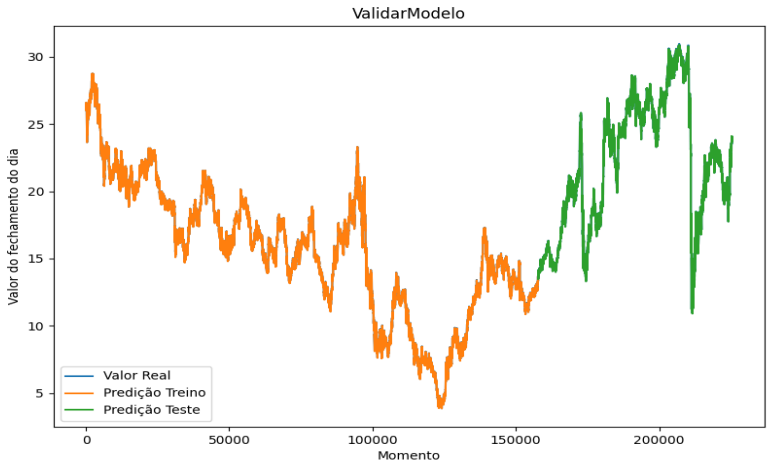
Fonte: Autoria Própria (2021)

Na Tabela 3 também é possível notar as taxas de lucros semelhantes, onde, mas diferente da Tabela 2, as taxas de lucros tiveram variações, onde foi o lucro na série real, seria sempre maior que a serie predita. Para realizar os testes referente a Tabela 3, foi utilizado a serie temporal do BBAS3 com intervalo de tempo de 5 minutos.

## 7. ANÁLISES E DISCUSSÕES

A resposta da LSTM, foi obtida através da melhor resposta em 10 testes. Logo, foram realizados 10 treinamentos da LSTM para cada série temporal. A LSTM teve resultado bastante satisfatório no ativo PETR4_M5, conforme mostra a Figura 17.

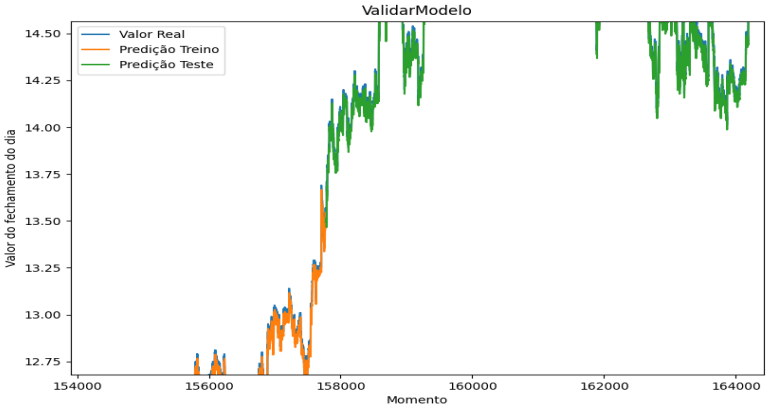
Figura 17 Treino e Teste LSTM para o ativo PETR4_M5




Fonte: Autoria Própria (2021)

É possível notar que na Figura 17 a predição da LSTM está tão bem, que não é possível observar a linha azul que representa o valor Real da série temporal. Para conseguir verificar o erro visualmente, foi necessário dar um zoom para ver a série Real. É possível ver através da Figura 18 essa proximidade do valor Real e da LSTM

Figura 18: Proximidade do valor real e do resultado da LSTM.



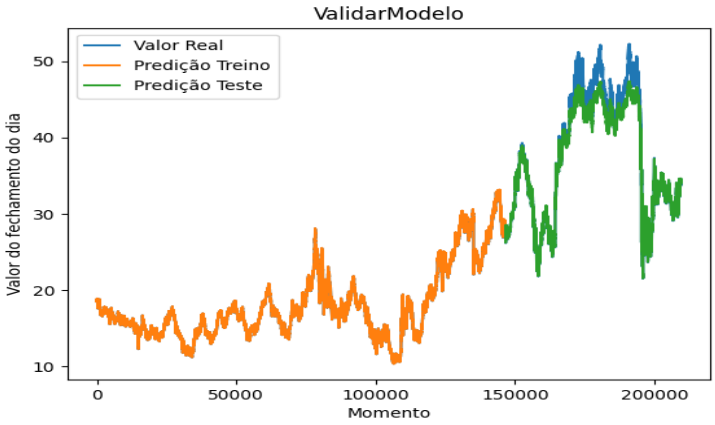

Fonte: Autoria Própria (2021)

É possível notar na Figura 18, o resultado da LSTM em relação a série temporal PETR4_M5 é bastante bom, visto que a previsão estava bem próxima ao valor real.

Ao realizar o mesmo teste no ativo BBAS3_M5, foi observado que enquanto a LSTM estava treinando ela obteve resultados bem parecidos com a PETR4_M5, mas, entretanto, quando efetuado a parte de teste a LSTM começou a errar bem mais em relação ao que errou na PETR4_M5.

É possível notar na Figura 19 3 a LSTM se saiu bem na parte de treino, mas na parte de testes ela não teve a mesma performance.

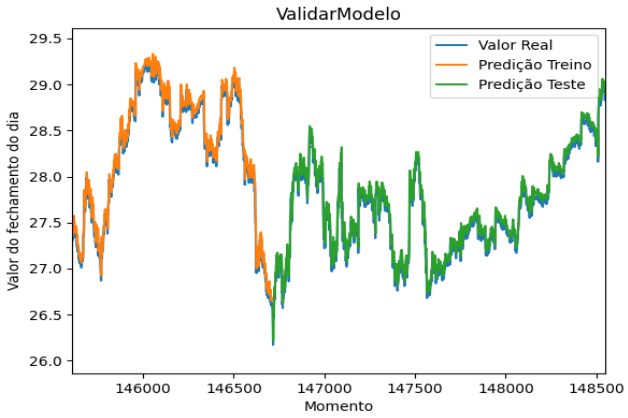
Figura 19: Treino e Teste LSTM para o ativo BBAS3_M5




Fonte: Autoria Própria (2021)

Ao realizar o mesmo procedimento da PETR4_M5. Ao dar um zoom na parte de treino é possível notar proximidade dos valores reais e dos valores preditos pela LSTM, mas ao fazer o mesmo procedimento na parte de teste, é possível notar a diferença entre os acertos.

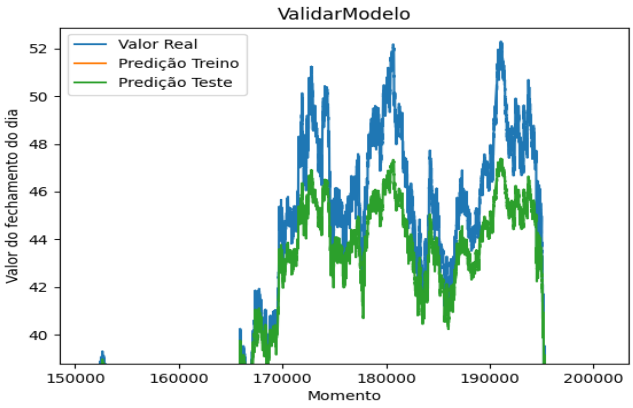
Figura 20: Proximidade do valor real e do resultado da LSTM.




Fonte: Autoria Própria (2021)

Inicialmente é possível verificar na Figura 20 que o desempenho na parte de teste está tão bom quanto na parte de treino, mas ao observar a fase de teste em um outro momento é possível verificar o aumento no erro da predição.

Figura 21: Proximidade do valor real e do resultado da LSTM.




Fonte: Autoria Própria (2021).

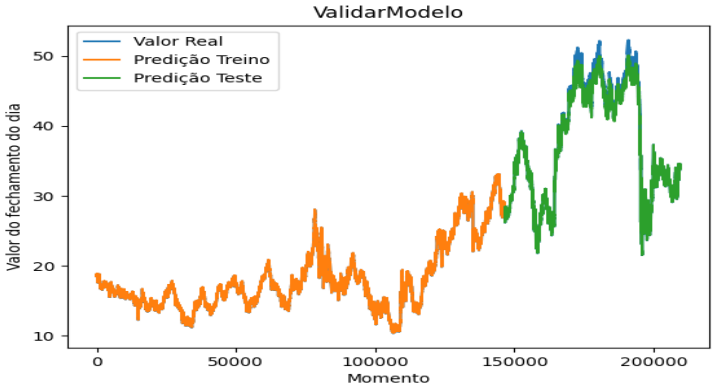
Visto que, a parte de treino teve a taxa de acerto bastante alta e enquanto na parte de teste teve a taxa de acerto menos expressiva, foi possível verificar um possível *overfitting* da rede LSTM. Então, foram realizadas alterações para corrigir esse problema. O *overfitting* é um problema que ocorre quando o modelo fica especialista somente no treino, assim ele "decora" o resultado do treino. Logo, quando ele vai para os testes ele não irá conseguir responder de maneira correta.

Foram utilizadas duas soluções, uma solução abordada foi reduzir o tempo de treinamento da rede de 100 épocas para 50 épocas e diminuir a quantidade de valores anteriores que a LSTM deve levar em consideração para prever o próximo valor, que inicialmente estava sendo utilizado em 2016 (equivale a uma semana de dados anteriores) e foi alterado para 1020. Para essa parametrização não foi adotado nenhum critério da literatura, mas sim testes arbitrários.

Essa solução teve um resultado superior à primeira tentativa, mas o resultado foi bastante parecido. Então, foi realizada uma nova solução. Onde a quantidade de épocas voltaria para 100, mas seria mantido o valor de quantidade de valores anteriores em 1020.

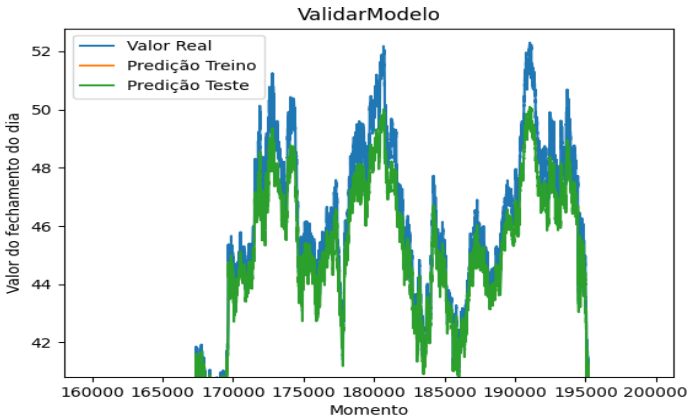
Nesta tentativa, a LSTM teve um resultado bastante superior, é possível observar na Figura 22 a melhoria na taxa de acerto da LSTM na parte de testes.

Figura 22: Treino e Teste LSTM para o ativo BBAS3_M5_R




Fonte: Autoria Própria (2021)

Figura 23: Proximidade do valor real e do resultado da LSTM BBAS3_M5_R.




Fonte: Autoria Própria (2021)

Ao comparar a Figura 21 e o Figura 23, é possível notar a melhora dos resultados gerados pela LSTM, visto que a LSTM começou a ter um resultado superior utilizando essa abordagem, então foi considerado a solução do *overfitting* que estava acontecendo.

## 8. CONSIDERAÇÕES FINAIS.

Foi possível aplicar a rede neural LSTM em ativos financeiros, analisar seus resultados e com isso criar um protótipo de uma ferramenta onde os investidores possam receber indicações sobre como ele deve agir ao investir naquele ativo.

A ferramenta web ainda está em desenvolvimento, mas já foi construído um MVP dela onde é possível fazer as operações básicas.

Este trabalho visou responder a seguinte questão: é possível uma rede neural ajudar os investidores do mercado de ações brasileira, com predições de ativos financeiros?

Observando os resultados obtidos, conclui-se que a LSTM consegue predizer os valores de fechamento dos ativos financeiros, e a utilização da ferramenta também auxiliará os investidores.

Durante o trabalho foi encontrado alguns problemas. Um era a lentidão de treinamento da LSTM quando utilizava ativos de 5min. Em um certo momento, o treinamento chegou a durar 8 horas, tendo em vista a demora foi utilizado um outro computador para tentar melhorar o tempo de treinamento, e com o novo computador o treinamento foi realizado em 5 horas.

Outro problema encontrado, foi o *overfitting* de alguns modelos, então, foram utilizadas técnicas para tratar esse problema, e a tratativa foi um sucesso, assim melhorando o desempenho do modelo.

Sugestões para trabalhos futuros:

- Tendo em vista o crescimento de investimento em criptomoedas, seria interessante verificar a predição desses ativos, trabalhando com a alta volatilidade do valor da moeda;

- Utilização de outros algoritmos de predição para comparar com a LSTM;

- Criar um produto para realizar investimentos reais. Após a validação da predição dos ativos, seria interessante transformar a ferramenta em uma aplicação de investimentos, que ao receber as devidas configurações essa ferramenta possa fazer investimentos de maneira automática.

## REFERÊNCIAS


ASSIS, C. A. S. DE. **Predição de Tendências em Séries Financeiras utilizando Meta-Classificadores**. [s.l.] CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS, 2019

Bao W, Yue J, Rao Y (2017) **A deep learning framework for financial time series using stacked autoencoders and long-short term memory**. PLoS ONE 12(7): e0180944. https://doi.org/10.1371/journal.pone.0180944

B3. B3 divulga estudo sobre os 2 milhões de investidores que entraram na bolsa entre 2019 e 2020. Disponível em: https://www.b3.com.br/pt_br/noticias/investidores.htm. Acesso em: 10 set. 2021.

**B3: entenda como funciona a bolsa de valores do Brasil e seus ativos: História da B3. História da B3.** Disponível em: https://www.onze.com.br/blog/b3/. Acesso em: 10 out. 2021.

F JUNIOR, Jose R. Redes Neurais Recorrentes — LSTM. Disponível em: https://medium.com/@web2ajax/redes-neurais-recorrentes-lstm-b90b720dc3f6os%20seguintes%20trabalhos.. Acesso em: 10 abr. 2021.

FISCHER, T.; KRAUSS, C. Deep learning with long short-term memory networks for financial market predictions. **European Journal of Operational Research**, v. 270, n. 2, p. 654–669, 2018. DOI: https://doi.org/10.1016/j.ejor.2017.11.054

GÉRON, Aurélien. **Mãos a Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow: conceitos, ferramentas e técnicas para a construção de sistemas inteligentes**. Rio de Janeiro: Alta Books, 2019. 554 p.

HAYKIN, Simon**. Redes Neurais: Princípios e Prática**. Porto Alegre: Bookman, 2001. 28 p.

Keras Documentação do Keras. Disponível em https:https://keras.io/

MARKOWITZ, H. Portfolio Selection. The Journal of Finance, v. 7, n. 1, p. 77–91, 1952. DOI: https://dx.doi.org/10.1111/j.1540-6261.1952.tb01525.

MOREIRA, Sandro. **Rede Neural Perceptron Adaline**. 2018. Disponível em: https://medium.com/ensina-ai/rede-neural-perceptron-adaline-8f69dc419d4e. Acesso em: 7 setembro. 2021.

NAMETALA, C. A. L. **Construção de um Robô Investidor baseado em Redes Neurais Artificiais e Preditores Econométricos**. [s.l.] Universidade Federal de Minas Gerais, 2017.

Numpy . Documentação do Numpy. Disponível em https: https://numpy.org/

PALMIERE, Sérgio Eduardo. Rede Perceptron de uma única camada. Disponível em: https://www.embarcados.com.br/rede-perceptron-de-uma-unica-camada/. Acesso em: 10 nov. 2021.Pandas Documentação do Pandas. Disponível em https: https://pandas.pydata.org/docs

PHI, Michael. Illustrated Guide to LSTM's and GRU's: A step by step explanation. Disponível em: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-

step-by-step-explanation-44e9eb85bf21. Acesso em: 08 out. 2021.

Python Software Foundation (2016). Documentação do Python 3.5.2. Disponível em https: https://www.python.org/psf/

SciKit-Learn Documentação do SciKit-Learn. Disponível em https:https://scikitlearn.org/stable/

Sepp Hochreiter, Jürgen Schmidhuber; **Memória longa de curto prazo**. Neural Comput 1997; 9 (8): 1735–1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735

GÉRON, Aurélien. **Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow**. 2. ed. Sebastopol: O'reilly, 2019. 286 p.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1069 ● Setor Universitário
Caixa Postal 86 ● CEP 74605-010
Goiânia ● Goiás ● Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br ● reitoria@pucgoias.edu.br

# RESOLUÇÃO n° 038/2020 – CEPE

## ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Luís Carlos Martins Arruda Junior do Curso de Ciência da Computação, matrícula 2017100780188-9, telefone: 62 99668 7476 e-mail lcmsjr.ceg@gmail.com, na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado Predição de comportamento de ações no mercado financeiro usando redes LSTM, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 15 de dezembro de 2021.

Assinatura do(s) autor(es): Luís Carlos M. A. Júnior

Nome completo do autor: Luís Carlos Martins Arruda Junior

Assinatura do professor-orientador: Max Gontijo

Nome completo do professor-orientador: Max Gontijo de Oliveira