PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Estudo de Algoritmos de Redes Neurais Recorrentes para Predição de Casos e Óbitos Diários pela COVID-19 no Centro-Oeste

PEDRO PAULO DE SOUSA COSTA

GOIÂNIA
2021

PEDRO PAULO DE SOUSA COSTA

Estudo de Algoritmos de Redes Neurais Recorrentes para Predição de Casos e Óbitos Diários pela COVID-19 no Centro-Oeste

Trabalho de Conclusão de Curso apresentado à Escola Politécnica, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Orientador:
Prof. Me. Aníbal Santos Jukemura

Banca Examinadora:
Profa. Ma. Lucília Gomes Ribeiro
Prof. Me. Gustavo Siqueira Vinhal

GOIÂNIA
2021

PEDRO PAULO DE SOUSA COSTA

Estudo de Algoritmos de Redes Neurais Recorrentes para Predição de Casos e Óbitos Diários pela COVID-19 no Centro-Oeste

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Ciência da Computação, e aprovado em sua forma final pela Escola Politécnica, da Pontifícia Universidade Católica de Goiás em 08/12/2021.

_____

Profa. Ma. Ludmilla Reis Pinheiro dos Santos
Coordenadora de Trabalho de Conclusão de Curso

Banca Examinadora:

_____

Orientador: Prof. Me. Aníbal Santos Jukemura

_____

Profa. Ma. Lucília Gomes Ribeiro

_____

Prof. Me. Gustavo Siqueira Vinhal

GOIÂNIA
2021

AGRADECIMENTOS

Primeiramente, agradeço aos meus pais, Neusa e Antônio, por sempre me apoiarem em relação aos estudos. Esses dois são feras demais.

A família Sousa e a família Costa, também sempre por me apoiaram em tudo na vida.

Ao amor da minha vida, Phelipe, que desde 11 de outubro de 2015 vem sendo a minha base para tudo. Te amo.

Ao meu orientador, Aníbal Santos Jukemura, pela orientação, tempo e dedicação compartilhadas comigo. E principalmente por não ter deixado eu desistir deste trabalho. Muito obrigado mesmo.

A professora Lucília Gomes Ribeiro e o professor Gustavo Siqueira Vinhal, que aceitaram participar da banca examinadora deste trabalho.

A Amanda, pela mega ajuda em dúvidas textuais e afins. Você me ajudou absurdamente.

Agradeço imensamente ao grupo Uma mesa e 6 cadeiras, pela essa trajetória acadêmica que enfrentemos juntos. E de nada por eu sempre passar os trabalhos para vocês.

Não posso deixar de agradecer o grupo OsMenó&Girls, porque se não eles choram, então muito obrigado por tudo.

RESUMO

Em razão das perdas e sequelas ocasionadas pelos casos e óbitos diários pela COVID-19 na região Centro-Oeste no Brasil, foram avaliados os modelos preditivos de Redes Neurais Recorrentes *Long Short Term Memory* e *Gated Recurrent Unit*, a partir de estudos obtidos em trabalhos relacionados, usando como base de dados as informações disponibilizadas no site Coronavírus Brasil (2021) pelo Ministério da Saúde. Utilizando-se de técnicas de pré-processamento para a otimização do conjunto de dados e usando como base os hiperparâmetros apresentados nos trabalhos relacionados, foi possível obter bons resultados para os dois modelos. O modelo *Long Short Term Memory* apresentou uma performance melhor do que o modelo *Gated Recurrent Unit*, em relação aos casos diários, tendo como resultado nas métricas de desempenho: *Mean Absolut Error* 963,92; *Root Mean Squared Error* 1261,53 e r2_*score* 0,94. Já o modelo *Gated Recurrent Unit* obteve melhores resultados para os óbitos diários, baseando-se os resultados nas métricas de desempenho: *Mean Absolut Error* 29,07; *Root Mean Squared Error* 40,10 e r2_*score* 0,96.

Palavras-chave: Coronavírus; análise preditiva; *deep learning*; lstm; gru.

ABSTRACT

Because of losts and sequels caused by the dacases and daily deaths as a result of COVID-19 disease placed on Midwest Region in Brazil, the predictive models of Recurrent Neural Networks, the Long Short Term Memory and Gated Recurrent Unit, were evaluated from studies obtained in related works, using as database the informations provided by the Coronavírus Brasil (2021) site from Ministry of Health. Using pre-processing techniques to optimize the dataset and using the hyperparameters presented in related works as a basis, it was possible to obtain good results for both models. The Long Short Term Memory model performed better than the Gated Recurrent Unit model, in relation to daily cases, resulting in the following performance metrics: Mean Absolut Error 963.92; Root Mean Squared Error 1261.53 and r2_score 0.94. On the other hand, the Gated Recurrent Unit model got better results for daily deaths, relying on the results in the performance metrics: Mean Absolut Error 29.07; Root Mean Squared Error 40.10 and r2_score 0.96.

Keywords: *Coronavirus, predictive analytics; Deep Learning; Long Short Term Memory; Gated Recurrent Unit Model.*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

DL	*Deep Learning*
GRU	*Gated Recurrent Unit*
IA	Inteligência Artificial
LSTM	*Long Short Term Memory*
MAE	*Mean Absolut Error*
ML	*Machine Learning*
MLP	*Multiple-Layer Perceptron*
OMS	Organização Mundial da Saúde
RMSE	*Root Mean Squared Error*
RNA	Rede Neural Artificial
RNR	Rede Neural Recorrente
SLP	*Single-Layer Perceptron*

SUMÁRIO

1 INTRODUÇÃO

A COVID-19 é uma doença viral, causada pelo SARS-CoV-2, o novo coronavírus, podendo provocar uma síndrome respiratória aguda grave, que se originou na cidade de Wuhan, China, no final de dezembro de 2019. No dia 28 de fevereiro de 2020, a Organização Mundial da Saúde (OMS) comunicou mais de 80 mil casos confirmados em todo o mundo, com menos de 2 meses desde a aparição do vírus. Em 11 de março de 2020, a OMS declarou o novo coronavírus como uma pandemia mundial (CUCINOTTA; VANELLI, 2020).

Segundo dados retirados da Johns Hopkins University (2021), no ano de 2020 foram registrados cerca de 83.632.587 casos confirmados e 1.880.668 óbitos por COVID-19 no mundo. Os três principais países mais afetados foram: Estados Unidos com 20.161.472 de casos e 351.754 de óbitos, Índia com 10.286.709 de casos e 194.949 de óbitos e Brasil com 7.675.973 de casos e 148.994 de óbitos.

Desde a declaração da OMS, várias medidas de prevenção, como testes rápidos, *lockdown*, uso de máscaras e distanciamento social, são aplicadas por diversos países para dificultar a propagação do vírus pelo mundo. Entretanto, apesar das prevenções aplicadas, a propagação da COVID-19 pelo mundo aconteceu rapidamente (ARUNKUMAR et al., 2021).

Em novembro de 2021, com aproximadamente 250.850.815 casos confirmados e 5.064.398 óbitos mundiais, calcula-se um aumento de 199,94% nos números de casos e 169,28% nos de óbitos, em relação a 2020 (Johns Hopkins University, 2021). Provavelmente esses números seriam maiores sem as medidas de prevenção.

O Brasil ocupa, em relação a casos e óbitos acumulados, o *top* três países mais afetados pelo vírus (Johns Hopkins University, 2021; World Health Organization 2021). Segundo os dados do Coronavírus Brasil (2021), o Brasil possui 21.897.025 de casos confirmados e 609.756 de óbitos pela COVID-19. Dividindo esses valores para cada região do país, das mais afetadas para as menos, temos: Sudeste (8.546.602 de casos e 289.865 de óbitos), Nordeste (4.868.694 de casos e 4.868.694 de óbitos), (Sul 4.261.97 7de e 96.085 de óbitos), Centro-Oeste

(2.349.413 de casos e 58.607 de óbitos) e Norte (1.870.339 de casos e 46.889 de óbitos).

Dado o cenário supracitado, considerando o prisma da pesquisa científica mundial, existem três comunidades científicas que contribuem significativamente para lidar com os problemas pandêmicos:

> (i) a comunidade de matemáticos aplicados, virologistas e epidemiologistas, desenvolvendo modelos de difusão sofisticados para as propriedades específicas de um determinado patógeno; (ii) a comunidade de cientistas de sistemas complexos que estudam a propagação de infecções usando modelos compartimentados, usando métodos e princípios da mecânica estatística e dinâmica não linear; e (iii) a comunidade de cientistas que incorporam inteligência artificial (IA) e, mais especificamente, abordagens de aprendizado profundo para produzir modelos preditivos precisos (RIBEIRO et al., 2020, tradução nossa).

Portanto, como descrito por Ribeiro et al. (2020), o uso de Inteligência Artificial (IA) resulta em contribuições significativas para predições de problemas emergentes durante a pandemia, como, o número de casos e óbitos diários por COVID-19 no Centro-Oeste do Brasil, foco principal deste trabalho.

Para que se explorasse esses temas, foram utilizados nesse trabalho, dois algoritmos de *Deep Learning* (DL), especificamente sobre Rede Neural Recorrente (RNR): o *Long Short Term Memory* (LSTM) e o *Gated Recurrent Unit* (GRU). Após o desenvolvimento dos modelos elaborados a partir dos algoritmos supracitados, foi apresentado o que obteve melhores resultados de predição, tendo como base, o conjunto de dados de estudo selecionado, as métricas *Root Mean Squared Error* (RMSE), *Mean Absolut Error* (MAE) e *r2_score*, que serão apresentados nos próximos capítulos.

1.2 Objetivo geral e específicos

Neste sentido, o objetivo geral explorado foi avaliar modelos preditivos de Rede Neural Recorrente, *Long Short Term Memory* e *Gated Recurrent Unit*, usando como base de dados informações do número de casos e óbitos diários por COVID-19 no Centro-Oeste do Brasil. Em decorrência deste trabalho, foi elaborada a

comparação dos resultados obtidos e com a análise da abordagem mais apropriada para predição do número de casos e óbitos diários estudados.

Além disso, os seguintes objetivos específicos e necessários foram adotados:

- Estudo e implementação de um mecanismo de limpeza da base de dados;
- Desenvolvimento dos modelos de predição citados, em ambiente de colaboração online;
- Análise das predições obtidas por meio de métricas de avaliação de desempenho.

1.3 Justificativa

Perante o apresentado até o momento, o presente trabalho justifica-se pelo próprio objetivo: avaliar os modelos preditivos *Long Short Term Memory* e *Gated Recurrent Unit*, a partir de estudos obtidos em trabalhos relacionados, usando como base de dados as informações disponibilizadas no site Coronavírus Brasil (2021) pelo Ministério da Saúde. Adotando-se os métodos estudados, foi possível encontrar o modelo que mais se adeque ao problema apresentado, com geração satisfatória de resultados, que serão apresentados no Capítulo 05.

1.4 Estrutura do Trabalho

O trabalho está organizado com a seguinte estrutura: O capítulo 2 descreve a revisão bibliográfica, apresentando as ideias e conceitos sobre Inteligência Artificial, *Machine Learning* (ML), *Deep Learning*, Rede Neural Artificial (RNA), Rede Neural Recorrente, *Long Short Term Memory* e *Gated Recurrent Unit*. No capítulo 3, o referencial teórico é apresentado, isto é, os trabalhos relacionados. O capítulo 4 descreve o ambiente de produção e metodologia utilizados. O capítulo 5 apresenta

os resultados obtidos, junto com as discussões deles. O capítulo 6, contém a conclusão do trabalho e sugestões para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, serão apresentados conceitos fundamentais para uma melhor compreensão das predições de casos e óbitos diários de COVID-19. A teoria estudada abrange conceitos de: Inteligência Artificial, *Machine Learning*, *Deep Learning*, Rede Neural Artificial, Rede Neural Recorrente, *Long Short Term Memory* e *Gated Recurrent Unit*.

A primeira definição a ser apresentada se relaciona a IA, exemplificando conceitos de sistemas e agentes. Em seguida, serão apresentadas as definições de *Machine Learning*, conceituando os tipos de aprendizagens e quais os tipos de problemas existentes nesse meio. Em terceiro, o conceito sobre *Deep Learning*, juntamente com RNA e RNR. Por fim, serão definidos os conceitos de LSTM e GRU.

2.1 IA

A definição de Inteligência Artificial baseia se por quatro sistemas, eles são: pensam como seres humanos, agem como seres humanos, pensam racionalmente e agem racionalmente (RUSSEL; NORVIG, 2013).

Nota se que existem sistemas humanos e racionais. Segundo, Russel e Norvig (2013):

> Uma abordagem centrada nos seres humanos deve ser em parte uma ciência empírica, envolvendo hipóteses e confirmação experimental. Uma abordagem racionalista envolve uma combinação de matemática e engenharia.

Dessa forma, este trabalho estabelece sistemas que agem racionalmente. Em qualquer sistema baseado na definição de IA, existem agentes responsáveis por realizar tarefas. Nos sistemas que agem racionalmente, existem os agentes racionais, que são responsáveis por alcançar o melhor resultado, ou, o melhor resultado esperado (RUSSEL; NORVIG, 2013).

Sistemas que agem racionalmente são a base para conceituação de *Machine Learning*, que será apresentada a seguir.

2.2 ML

Machine Learning, ou, Aprendizado de Máquina, pode ser definido como a habilidade um agente racional de identificar padrões em um conjunto de informações, e dessa forma obter conhecimento. Também pode ser descrita como uma área de pesquisa computacional, que visa estudar um conjunto de métodos que identificam automaticamente padrões em um conjunto de dados. E usar os padrões encontrados para predizer eventualidades futuras (KEVIN P. MURPHY, 1988; SILVA, 2020).

Russell e Norvig (2013) descrevem que existem três tipos de Aprendizagem de Máquina, elas são:

- Aprendizagem não supervisionada: não é fornecido nenhum conjunto de informação de entrada para o sistema, ele aprende sozinho as entradas. A tarefa mais usual dessa aprendizagem é o agrupamento, onde é realizada a identificação de grupos de entradas úteis.
- Aprendizagem supervisionada: o sistema recebe um conjunto de informação em pares, compostos por entrada e saída, e dessa forma aprendem uma função mapeadora que seja capaz de aproximar os dados de entrada com os dados de saída.
- Aprendizagem por reforço: o sistema é apenas informado quando o conjunto de informação está errado, a informação correta não é fornecida a ele. O sistema tem que aprender sozinho através de recompensas ou punições.

Neste trabalho será usada a aprendizagem supervisionada. Nesse tipo de aprendizagem, existe a função mapeadora, que também é conhecida como hipótese. Essa aprendizagem tem como finalidade escolher um modelo que estabelece o domínio da hipótese, e uma técnica de otimização capaz de encontrar a hipótese mais apropriada para o domínio. A otimização tem como objetivo minimizar uma função de perda, e por consequência a taxa de erro. A taxa de erro sinaliza quanto longe a hipótese está próxima da solução ideal (SILVA, 2020).

O conjunto de informações supracitado possui valores que podem ser divididos como problemas de regressão ou classificação. Quando os valores deste conjunto apresentarem dados como, dia ensolarado, nublado, chuvoso, tipos de plantas etc, valores que possam ser divididos em grupos, o problema é chamado de classificação. Quando esses valores forem temporais, como, temperatura diária, o preço semanal da gasolina etc, o problema é chamado de regressão (RUSSELL; NORVIG, 2013).

O problema do conjunto de informações usadas neste trabalho é considerado de regressão, pois as características dos seus dados são temporais. Características essas que são, casos e óbitos diários de COVID-19 no Centro-Oeste.

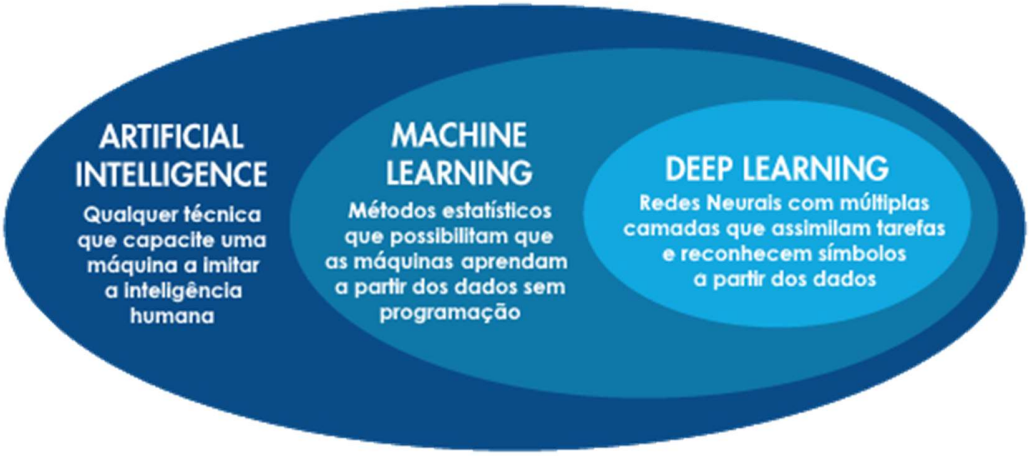
Deep Learning é uma subárea do *Machine Learning*, e será descrito a seguir.

2.3 DL

Em *Machine Learning* as características dos dados são fornecidas de forma manual, porque o modelo não consegue fazer a extração deles. Para fazer essa extração das características dos dados de forma automática, usa-se aprendizagem por representação. Aprendizagem por representação consiste em conceder uma grande quantidade de exemplos para o modelo, assim, identificando padrões. *Deep Learning* é um exemplo de aprendizagem por representação (SILVA, 2020).

Segundo GAO; WANG, 2019, DL permite que os computadores criem conceitos complexos baseando-se em conceitos mais simples. Em outra concepção, *Machine Learning* permite que um computador aprenda um programa a partir de várias etapas. Sendo que cada etapa pode ser considerada um estado de memória do computador, após a execução de um conjunto de instruções.

Figura 1 – Relação entre IA, ML e DL.



Fonte: Lavagnoli (2019).

A Figura 1 representa a relação entre Inteligência Artificial, *Machine Learning* e *Deep Learning*, e descreve rapidamente os seus conceitos. De acordo com GAO; WANG, 2019, DL é um tipo de ML, que possui um grande poder ao aprender representando o mundo a partir de conceitos. Com cada conceito definido a partir de um conceito mais simples, dessa forma, realizando representações mais abstratas em termos de outras menos abstratas.

DL tem como estrutura Rede Neural Artificial, que será descrita a seguir.

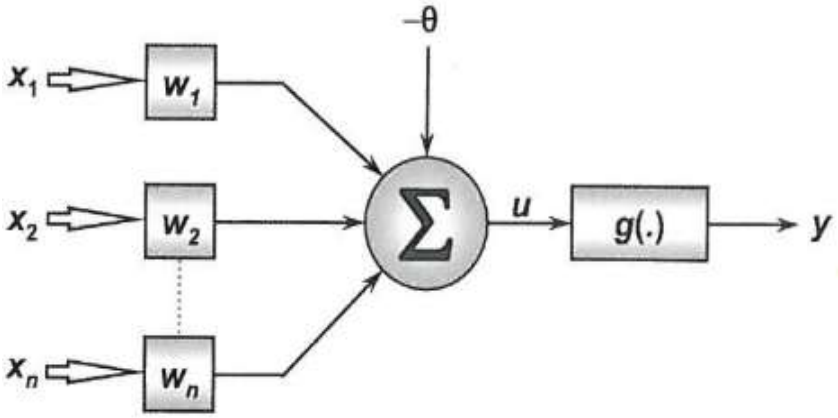
2.3.1 RNA

Para a produção de máquinas mais inteligentes, um exemplo de modelo a ser seguido é o cérebro humano. O sistema nervoso humano serviu como ideia para o desenvolvimento das Redes Neurais Artificiais, com propósito de simular a habilidade de aprendizagem humana na obtenção de conhecimento (FACELI et al., 2021).

Uma RNA é um sistema computacional, desenvolvido a partir de um conjunto de unidades conectadas, que executam tarefas simples. As unidades são

chamadas de neurônios e calculam funções matemáticas (FACELI et al., 2021; RUSSELL; NORVIG, 2013). A Figura 2 mostra um neurônio artificial.

Figura 2 – Representação de um neurônio artificial.



Fonte: SILVA; SPATTI; FLAUZINO, 2010.

Matematicamente a Figura 2 pode ser representada:

$$u = \sum_{i=1}^{n} w_i x_i - \theta \quad (1)$$

$$y = g(u) \quad (2)$$

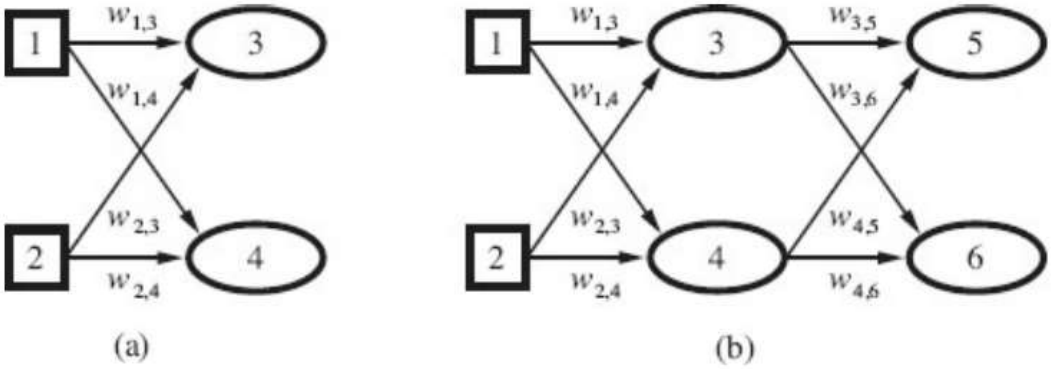
Segundo SILVA; SPATTI; FLAUZINO (2010), usando como base a Figura 2, um neurônio artificial pode ser dividido em:

1. Sinais de entrada: representados por x_1, x_2 e x_n, são os dados recebidos para a aprendizagem dos algoritmos. Em geral, são usados vetores com os dados normalizados como entrada;

2. Pesos sinápticos: descritos pelos símbolos w_1, w_2 e w_n, servem para ponderar cada um dos sinais de entrada. Com isso, qualificando sua importância a funcionalidade do neurônio;
3. Combinador linear: retratado pelo símbolo do somatório ($\sum$), seu objetivo é juntar todos os sinais de entrada já ponderados para gerar um valor de potencial de ativação;
4. Limiar de ativação: simbolizado pelo "teta" (θ), tem como função definir se o resultado gerado pelo combinador linear, gera um valor apropriado para saída do neurônio;
5. Potencial de ativação: representado pelo símbolo u. É a diferença entre o combinador linear e o limiar de ativação. Se o valor é $u \geq 0$, o neurônio gera um potencial excitatório, caso contrário, um potencial inibitório;
6. Função de ativação: simbolizada pela letra g, a sua função é delimitar as saídas do neurônio, isso é, dentro de um intervalo de valores;
7. Sinal de saída: retratada pelo símbolo y. É o valor final produzido por sinais de entrada.

De acordo com SILVA (2020), RNA são arquiteturas estruturadas em camadas, sendo que existem duas principais: RNA simples, conhecida como Perceptron de Única Camada (SLP – *Single-Layer* Perceptron), contendo uma camada de entrada e uma camada de saída. RNA profunda, conhecida como Perceptron de Múltiplas Camadas (MLP – *Multiple-Layer* Perceptron). Diferente da SLP, a MLP possui uma complexidade maior, pois entre suas camadas de entrada e saída existem as camadas ocultas. As camadas ocultas são as responsáveis por aprender as informações dos sinais de entrada. A Figura 3 mostra visualmente a diferença entre uma SLP e MLP.

Figura 3 – Representação de uma SLP e MLP.

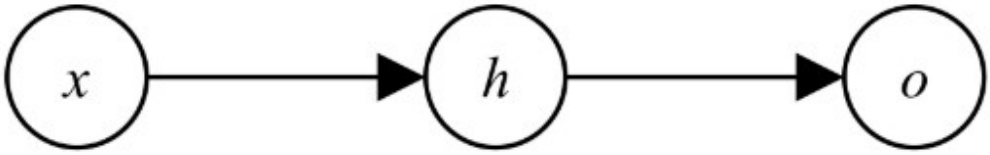


Fonte: RUSSELL; NORVIG (2013).

A SLP é representada com duas unidades de entrada e duas unidades de saída (Figura 3a). Já a MLP é representada com duas unidades de entrada, duas unidades de saída e entre elas uma camada oculta com duas unidades (Figura 3b) (RUSSELL; NORVIG, 2013).

Os dois principais tipos de RNA são: Rede Neural *Feed-Forward* que tem apenas uma unidade conectada com suas unidades anteriores (SILVA, 2020) (Figura 4) e Rede Neural Recorrente que será abordado no próximo tópico.

Figura 4 – Exemplificação de uma Rede Neural *Feed-Forward*.

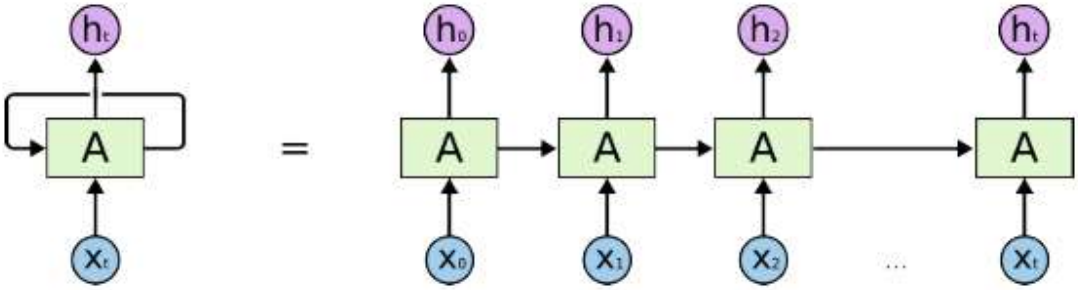


Fonte: SILVA (2020).

2.3.2 RNR

Conforme descrito no tópico anterior, pode-se afirmar que uma Rede Neural Recorrente é uma Rede Neural Artificial recursiva, em que os sinais de entrada são dependentes das etapas que ocorreram nos neurônios passados (MARUMO, 2018; VASCO, 2020). A Figura 5 mostra de forma exemplificada uma RNR.

Figura 5 – Exemplificação de uma RNR.



Fonte: MARUMO (2018).

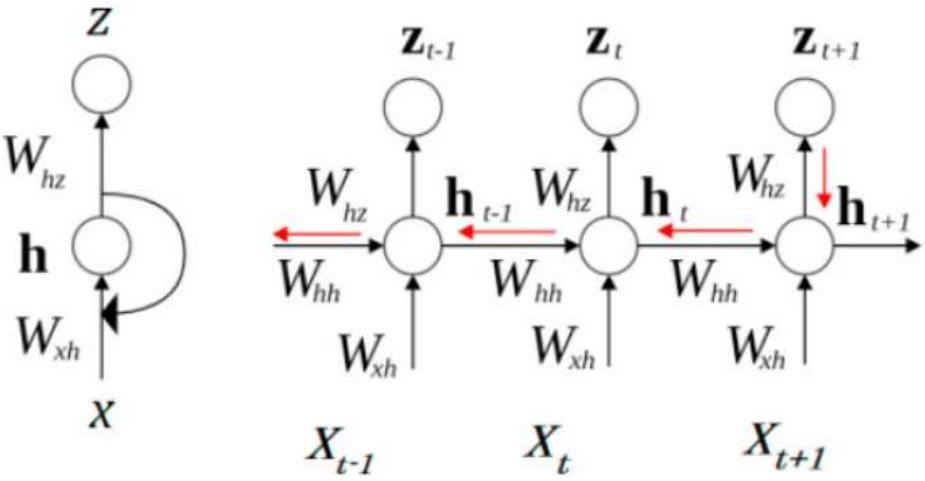
A entrada A recebe os sinais de entrada x e produzem a saída h, que são usadas para retroalimentação posteriores das camadas ocultas (Figura 5). Com a alimentação de saída do estado anterior, h_{t-1}, na etapa atual, h_t, forma uma representação de memória, pois os dados do passado são usados no estado presente, criando assim uma dependência sequencial dos dados (MARUMO, 2018).

Outra forma de descrever uma RNA é saber que ela possui duas entradas de dados diferentes: uma que está no presente e outra que está no passado. As duas entradas são usadas para gerar uma nova entrada de dados, isto é feito por meio do *feedback*, no qual a saída de cada momento é usada como entrada para o próximo instante. Assim, as RNRs são consideradas mais parecidas com a forma de processamento de informações dos seres humanos (VASCO, 2020).

De acordo com VASCO (2020), a estrutura de uma RNR pode ser dividida em três partes: camada de entrada, camada oculta e camada de saída. Sendo que

os neurônios da camada de entrada são conectados com os da camada oculta e, por fim, conectados à camada de saída.

Figura 6 – Representação matemática de uma RNR.



Fonte: VASCO (2020).

Segundo VASCO (2020), na Figura 6 pode-se representar a equação matemática de entrada das camadas ocultas como

$$h_t = g_n(w_{xh}x_t + w_{hh}h_{t-1} + b_h) \quad (3)$$

, onde:

- h_t representa a camada oculta no instante t;
- g_n é a função de ativação;
- w_{xh} é a matriz de pesos de entrada;
- x_t a entrada no instante t;
- h_{t-1} a camada oculta no instante $t-1$;

- w_{hh} a matriz de peso do neurônio recorrente;
- b_h o valor do viés.

A saída da camada oculta é representada por

$$z_t = g_n(w_{hz}h_t + b_z) \quad (4)$$

, onde:

- z_t representa o vetor de saída;
- w_{hz} a matriz de peso para a camada de saída;
- b_z o viés.

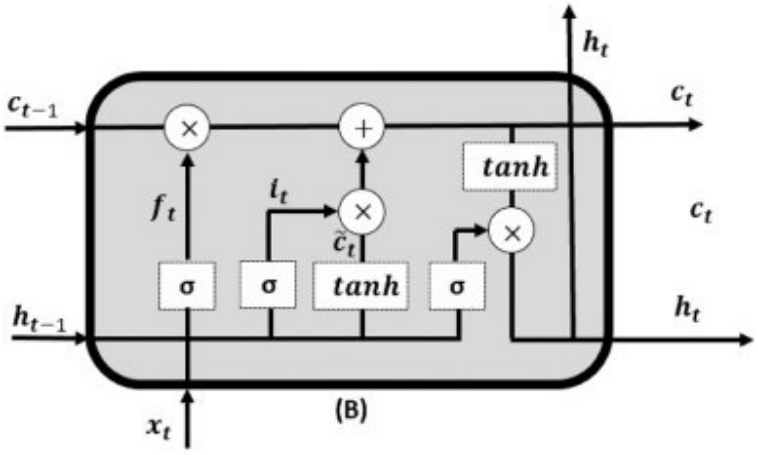
Um dos problemas da RNR é a dissipação do gradiente, isso faz com que o gradiente da função de custo diminua exponencialmente conforme o tempo e dessa maneira o modelo para de aprender (VASCO, 2020). Um algoritmo que corrige esse problema é o *Long Short Term Memory.*

O entendimento de uma RNR é fundamental para entender o funcionamento de uma LSTM e *Grated Recurrent Unit*.

2.3.2.1 LSTM

A rede *Long Short Term Memory* é um tipo de Rede Neural Recorrente, que possui a habilidade de aprender dependências de longo prazo. Ela foi proposta por Hochreiter e Schmidhuber, para solucionar o problema de dissipação do gradiente, também conhecido como explosão do gradiente (ARUNKUMAR et al., 2021; VASCO, 2020). A Figura 7 mostra uma célula LSTM.

Figura 7 – Célula LSTM.



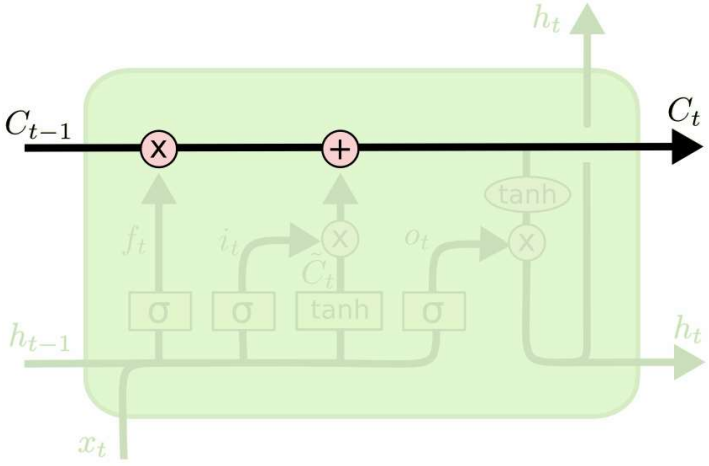
Fonte: ARUNKUMAR et al. (2021).

Uma célula LSTM (Figura 7) pode ser dividida em cinco estados: *cell state*, *forget gate*, *input gate*, *cell update* e *output gate*. Cada uma dessas partes será descrita a seguir, conforme descrito por MARUMO (2018) e VASCO (2020).

Cell state (C_t) é responsável por levar as informações da célula anterior para a célula atual (Figura 8). Sua equação matemática é representada por

$$c_t = F_t \, X \, c_{t-1} + I_t \, X \tanh\left(\mathrm{w_{xc}x_t} + \mathrm{w_{hc}h_{t-1}} + \mathrm{b_c}\right) \quad (5)$$

Figura 8 – Estado *cell state*.



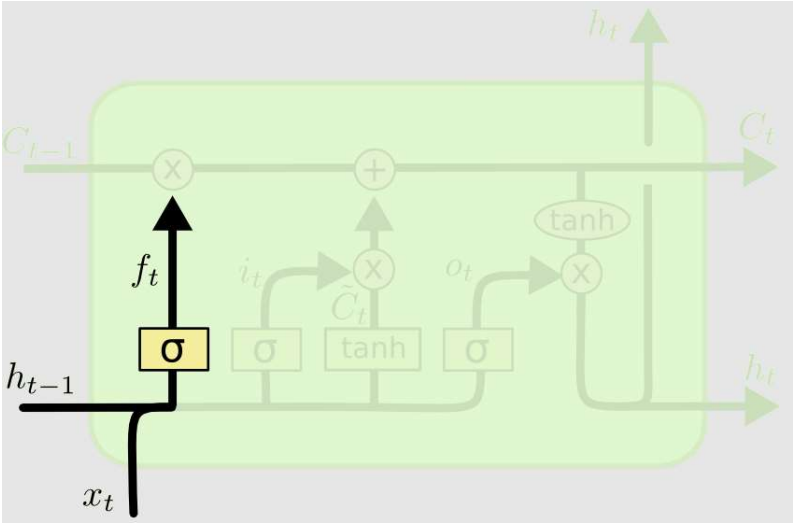
Fonte: JUNIOR, 2019

Forget gate (f_t) é responsável por guardar, ou, descartar informações do estado anterior e isso se deve a sua equação matemática (Figura 9)

$$f_t = \sigma(w_{xf}x_t + w_{hf}h_{t-1} + w_{cf}c_t - 1 + b_f) \quad (6)$$

Os valores possíveis para F_t dependem de uma função sigmoide, que varia entre 0 e 1, caso for 0 as informações do estado anterior são esquecidas, caso contrário, as informações são mantidas.

Figura 9 – Estado *forget gate*.



Fonte: JUNIOR, 2019.

Input gate (i_t) tem como objetivo definir o que vai ser inserido no *cell state* (Figura 10). Isso ocorre através de uma multiplicação entre uma camada sigmoide, valores variam entre 0 e 1, e uma camada de tangente hiperbólica, valores variam entre -1 e 1. Matematicamente o *input gate* é representado por

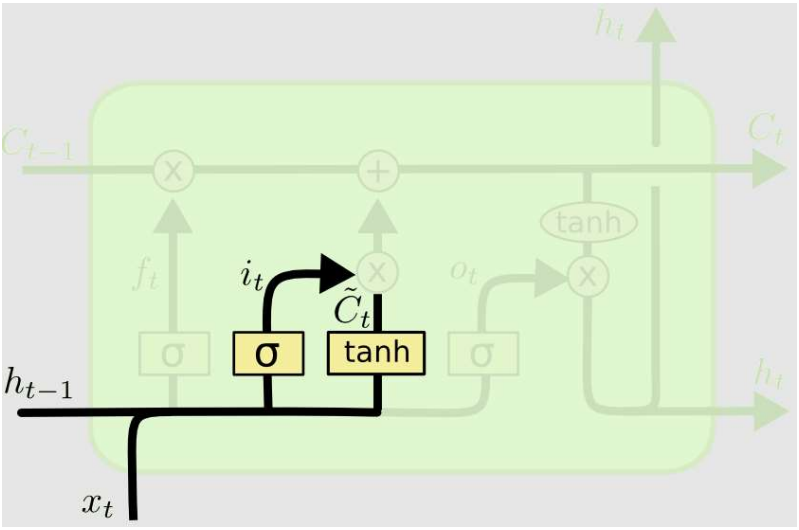
$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (7)$$

, caso o resultado desta sigmoide for 1 o valor de $\tilde{C}_t$, dado pela fórmula

$$\tilde{c} = \tanh\left(W_c[h_{t-1}, x_t] + b_c\right) \quad (8)$$

entra no *cell state*, caso contrário, não entram.

Figura 10 – Estado *input gate*.

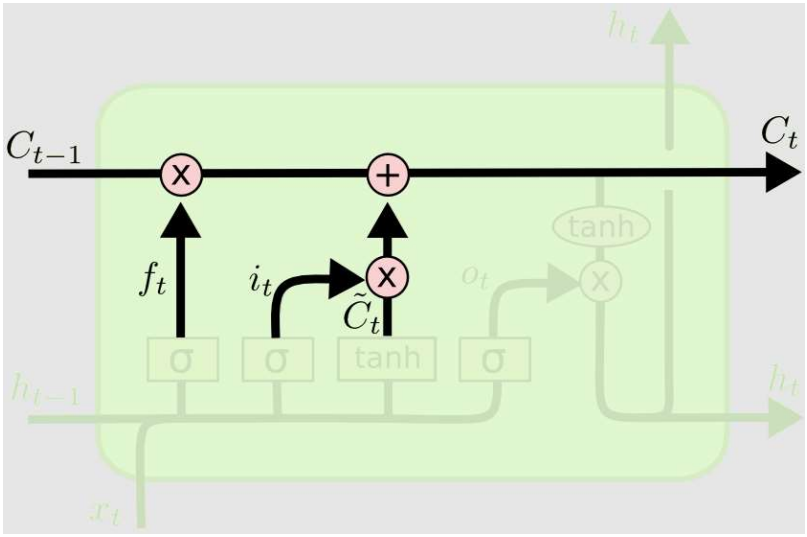


Fonte: JUNIOR, 2019.

Cell update (C_t) é a atualização da *cell state* do estado anterior (Figura 11). Essa atualização é feita através da multiplicação do *forget gate* com C_{t-1} e com a soma do *input gate*. Matematicamente representada por

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (9)$$

Figura 11 – Estado *cell update*.

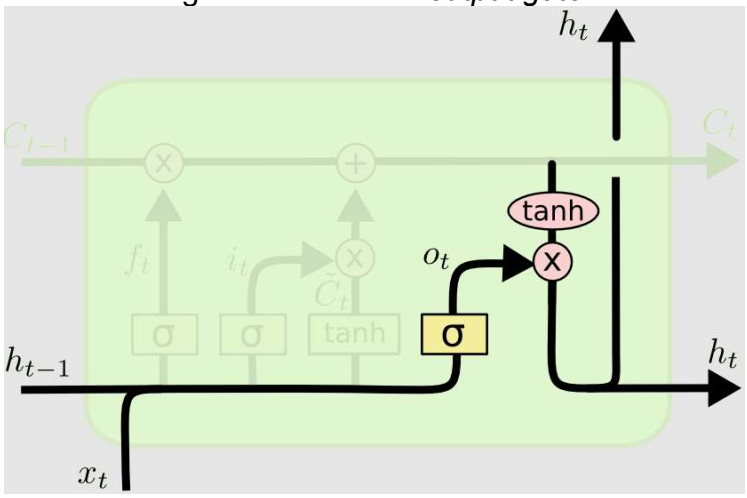


Fonte: JUNIOR, 2019.

Output gate (O_t) determina qual informação nova será passada para o próximo tempo e para a saída da rede (Figura 12). Sua representação matemática é dada por

$$O_t = \sigma(W_{xo}X_t + W_{h0}h_{t-1} + W_{Co} + b_0) \tag{10}$$

Figura 12 – Estado *output gate*.

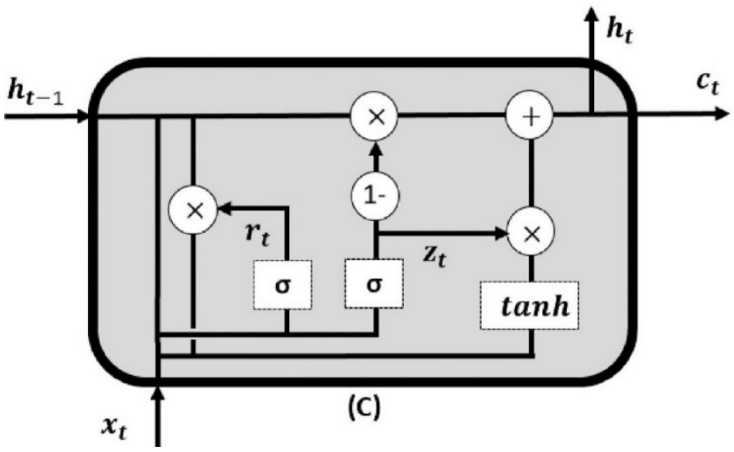


Fonte: JUNIOR, 2019.

2.3.2.2 GRU

A rede *Grated Recurrent Unit* é um tipo de Rede Neural Recorrente semelhante a *Long Short Term Memory*. Entretanto, essa RNR possui sua estrutura mais simples (Figura 13). Diferente da LSTM, a GRU tem apenas dois tipos de estados: *update gate* e *reset gate*. Com sua estrutura mais simples, acaba contendo menos parâmetros, seu processo de treino é mais rápido e apresenta resultados melhores com um menor número de dados menor, diferente da LSMT, que apresenta um melhor resultado com um maior número de dados (VASCO, 2020).

Figura 13 – Célula GRU.



Fonte: ARUNKUMAR et al. (2021)

A partir da Figura 7 e em comparação ao que foi explica sobre a célula LSTM, a GRU tem as seguintes diferenças: o *input gate* e o *forget gate* são unidos em apenas um estado chamado de *update gate* (Figura 14) e não existe mais o estado *cell state* (ARUNKUMAR et al., 2021; VASCO, 2020).

De acordo com VASCO (2020) e ARUNKUMAR et al. (2021), as equações matemáticas de uma célula GRU são:

$$Z_t = \sigma(W_x X_t + U_Z h_{t-1} + b_z) \quad (11)$$

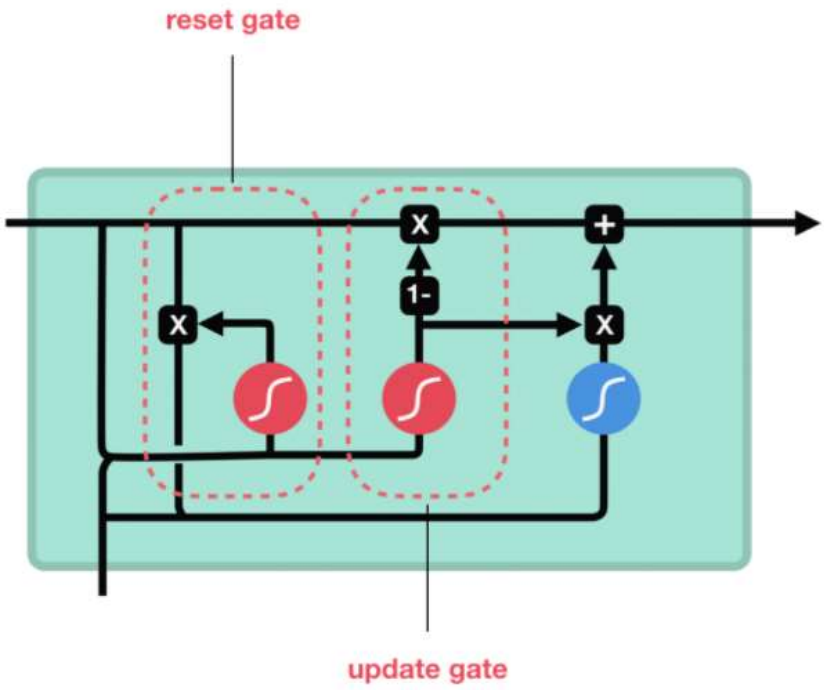
$$R_t = \sigma(W_R x_t + U_R h_t - 1 + \ b_R) \quad (12)$$

$$\hat{h}_t = tang(W_h X_t + U_h(R_t h_t - 1) + b_h) \quad (13)$$

$$h_t = (1 - Z_t)h_t - 1 + Z_t \hat{h}_t \quad (14)$$

Onde Z_t é o *update gate*, R_t *reset gate*, $\hat{h}_t$ vetor candidato a ativação e h_t vetor de saída.

Figura 14 – *Update gate* e *reset gate* GRU.



Fonte: ARUNKUMAR et al. (2021)

Desde a exemplificação dos conceitos de Inteligência Artificial até como são estruturadas as células de RNR LSMT e GRU, existem trabalhos na área que mostram várias maneiras de realizar predições no número de casos e óbitos por COVID-19, seja por dados acumulativos ou diários. Dos estudos existentes, foram selecionados dois trabalhos que serviram como base para o desenvolvimento deste. O próximo capítulo tem como finalidade a descrição desses trabalhos.

3 TRABALHOS RELACIONADOS

A pandemia da COVID-19 fez o número de estudos voltados para área de predição com Redes Neurais Recorrentes se intensificassem. Com objetivo de realizar predições de sessenta dias nos casos confirmados cumulativos, casos recuperados cumulativos e óbitos cumulativos por COVID-19, nos dez principais países mais afetados pelo vírus, ARUNKUMAR et al. (2021), utilizaram de RNRs com algoritmos de *Gated Recurrent Unit* e *Long Short Term Memory*.

De acordo com ARUNKUMAR et al. (2021), os modelos foram desenvolvidos na plataforma *Google COLAB*, com o uso da biblioteca *PyTorchdeep* e o conjunto de dados disponibilizados por *Johns Hopkins University*.

Os dez principais países mais afetados foram: Estados Unidos, Brasil, Índia, Rússia, África do Sul, México, Peru, Chile, Reino Unido e Irã (ARUNKUMAR et al., 2021).

Para a normalização do conjunto de dados, ARUNKUMAR et al. (2021), usou o modelo *MinMaxScaler* para ajustar os valores do conjunto entre 0 e 1 e, desta forma, facilitar a realização da fase de treino. O conjunto de dados não foi especificado pelo autor, isto é, não foi informada a quantidade de registros usados. Porém, a divisão do conjunto de dados ficou definida como os 14 últimos registros para teste e o restante menos os últimos 14 dias ficaram para treino. O passo do tempo usado foi de comprimento 30, isso significa que foram usados 30 registros para fazer a predição do registro 31 e assim por diante.

Com o intuito de construir um modelo personalizado para cada um dos dez países mais afetados pela COVID-19, ARUNKUMAR et al. (2021) usaram as melhores combinações de hiperparâmetros. Foram definidos os hiperparâmetros de número de nós em cada camada (10, 100, 200, 300), número de camadas ocultas (1, 2, 3, 4, 5), taxa de aprendizagem (0.1, 0.01, 0.001, 0.0001, 0.00001) e número de épocas. Os valores que estão entre parênteses são os intervalos dos hiperparâmetros que foram testados, para encontrar o melhor modelo por país. Além disso, usou-se como otimizador o algoritmo Adam, para balancear os pesos da rede com uma função de perda. Adam é um algoritmo de otimização que é

responsável por balancear os pesos da rede com uma função de perda. A função de perda usada foi erro quadrático médio.

As métricas de desempenho usadas para encontrar o melhor modelo definidas por ARUNKUMAR et al. (2021) foram o erro quadrático médio e a raiz quadrada do erro médio. Segundo os autores, quanto menor os valores das métricas, melhor é o desempenho do modelo.

Dessa forma, ARUNKUMAR et al. (2021), conseguiram aplicar esses hiperparâmetros para cada país e obteve os resultados apresentados nas figuras 15, 16 e 17, ilustrando casos confirmados cumulativos, casos recuperados cumulativos e óbitos cumulativos por COVID-19, com base nas métricas de desempenho supracitadas.

Todos os resultados dos modelos estão na escala de expoente de dez positivo (E+) para melhor visualização. Para os casos confirmados cumulativos (Figura 15), o modelo LSTM apresentou um melhor desempenho nos Estados Unidos, Brasil, África do Sul, Peru, Chile e Irã. Já o modelo GRU apresentou um melhor desempenho na Índia, Rússia, México e Reino Unido. Para os casos recuperados cumulativos (Figura 16), o LSTM foi melhor na Índia, África do Sul, Chile, Reino Unido e Irã, enquanto que o modelo GRU foi melhor nos USA, Brasil, Rússia, Mexico e Peru. Por fim, para os óbitos cumulativos (Figura 17) o modelo LSTM teve um melhor desempenho no Brasil, Índia, Rússia, África do Sul, México e Irã, já o modelo GRU foi melhor nos Estados Unidos, Peru, Chile e Reino Unido.

Foi notado que quantitativamente, o modelo LSTM foi 17 vezes mais apropriado que o modelo GRU, sendo, seis vezes nos casos confirmados cumulativos, cinco vezes nos casos recuperados cumulativos e seis vezes nos óbitos cumulativos. Qualitativamente, o modelo LSTM se destaca novamente, pois apresenta valores menores que o modelo GRU de acordo com as métricas de desempenho.

As Figuras 15, 16 e 17 possuem as mesmas colunas e estão divididas como: “No.” - número de cada país; “*Country*” - nome em Inglês dos dez países mais afetados; “*Epochs*” - número de épocas, “*Hidden size*” - número de nós em cada camada; “*Number of layers*” - número de camadas ocultas; “*Learning rate*” - taxa de

aprendizagem, “MSE” - sigla em inglês *Root Mean Squared Error*, em Português Erro Quadrático Médio; “RMSE” - sigla em Inglês para *Root Mean Squared Error*, em Português Raiz Quadrada do Erro Médio.

Figura 15 – Casos confirmados cumulativos e seus resultados

No.	Country	RNN model	Epochs	Hidden size	Number of layers	Learning rate	MSE	RMSE
1	USA	GRU	5.00E+02	3.00E+02	2.00E+00	1.00E-05	2.96E+12	1.72E+06
		LSTM	1.69E+03	3.00E+02	3.00E+00	1.00E-05	2.87E+12	1.69E+06
2	Brazil	GRU	3.00E+02	3.00E+02	2.00E+00	1.00E-05	1.86E+10	1.36E+05
		LSTM	1.00E+02	3.00E+02	2.00E+00	1.00E-05	1.76E+10	1.33E+05
3	India	GRU	1.00E+02	3.00E+02	2.00E+00	1.00E-05	4.58E+08	2.14E+04
		LSTM	1.80E+03	3.00E+02	2.00E+00	1.00E-05	5.57E+08	2.36E+04
4	Russia	GRU	1.00E+03	3.00E+02	2.00E+00	1.00E-05	8.79E+05	9.37E+02
		LSTM	2.56E+02	3.00E+02	2.00E+00	1.00E-05	1.10E+06	1.05E+03
5	South Africa	GRU	1.52E+03	3.00E+02	2.00E+00	1.00E-05	1.08E+07	3.29E+03
		LSTM	2.00E+03	3.00E+02	2.00E+00	1.00E-05	3.64E+07	6.03E+03
6	Mexico	GRU	5.00E+02	3.00E+02	2.00E+00	1.00E-05	1.60E+07	4.00E+03
		LSTM	1.50E+03	3.00E+02	2.00E+00	1.00E-05	2.36E+07	4.86E+03
7	Peru	GRU	6.00E+02	3.00E+02	2.00E+00	1.00E-05	5.37E+07	7.33E+03
		LSTM	7.00E+01	3.00E+02	2.00E+00	1.00E-05	1.97E+07	4.44E+03
8	Chile	GRU	3.00E+03	3.00E+02	2.00E+00	1.00E-05	6.52E+06	2.55E+03
		LSTM	1.30E+03	3.00E+02	2.00E+00	1.00E-05	1.49E+06	1.22E+03
9	UK	GRU	2.50E+02	3.00E+02	2.00E+00	1.00E-05	1.77E+05	4.21E+02
		LSTM	3.00E+02	3.00E+02	2.00E+00	1.00E-05	3.84E+05	6.91E+02
10	Iran	GRU	4.50E+02	3.00E+02	2.00E+00	1.00E-05	3.06E+05	5.52E+02
		LSTM	7.05E+02	3.00E+02	2.00E+00	1.00E-05	1.78E+04	1.33E+02

Fonte: ARUNKUMAR et al. (2021).

Figura 16 – Casos recuperados cumulativos e seus resultados

No.	Country	RNN model	Epochs	Hidden size	Number of layers	Learning rate	MSE	RMSE
1	USA	GRU	2.50E+02	3.00E+02	2.00E+00	1.00E-05	3.61E+11	6.01E+05
		LSTM	2.30E+01	3.00E+02	3.00E+00	1.00E-05	3.72E+11	6.10E+05
2	Brazil	GRU	6.56E+02	3.00E+02	2.00E+00	1.00E-05	8.53E+09	9.24E+04
		LSTM	4.02E+02	3.00E+02	2.00E+00	1.00E-05	1.44E+12	1.20E+06
3	India	GRU	1.52E+03	3.00E+02	2.00E+00	1.00E-05	7.67E+04	2.76E+02
		LSTM	1.26E+03	3.00E+02	2.00E+00	1.00E-05	6.62E+04	2.57E+02
4	Russia	GRU	3.00E+02	3.00E+02	2.00E+00	1.00E-05	1.17E+06	1.08E+03
		LSTM	1.00E+02	3.00E+02	2.00E+00	1.00E-05	7.65E+06	2.77E+03
5	South Africa	GRU	2.70E+03	3.00E+02	2.00E+00	1.00E-05	1.67E+07	4.08E+03
		LSTM	2.00E+03	3.00E+02	2.00E+00	1.00E-05	2.15E+06	4.05E+03
6	Mexico	GRU	6.50E+02	3.00E+02	2.00E+00	1.00E-05	1.61E+08	1.27E+04
		LSTM	1.65E+03	3.00E+02	2.00E+00	1.00E-05	1.73E+08	1.32E+04
7	Peru	GRU	2.66E+02	3.00E+02	2.00E+00	1.00E-05	6.56E+06	2.56E+03
		LSTM	7.50E+01	3.00E+02	2.00E+00	1.00E-05	2.13E+07	4.61E+03
8	Chile	GRU	3.00E+02	3.00E+02	2.00E+00	1.00E-05	1.32E+06	1.15E+03
		LSTM	2.60E+02	3.00E+02	2.00E+00	1.00E-05	7.65E+05	8.74E+02
9	UK	GRU	3.00E+02	3.00E+02	2.00E+00	1.00E-05	1.19E+01	3.40E+00
		LSTM	6.46E+02	3.00E+02	2.00E+00	1.00E-05	9.20E+00	3.03E+00
10	Iran	GRU	7.00E+02	3.00E+02	2.00E+00	1.00E-05	1.09E+06	1.04E+03

Fonte: ARUNKUMAR et al. (2021).

Figura 17 – Óbitos cumulativos e seus resultados

No.	Country	RNN model	Epochs	Hidden size	Number of layers	Learning rate	MSE	RMSE
1	USA	GRU	5.00E+02	3.00E+02	2.00E+00	1.00E-05	2.09E+05	4.57E+02
		LSTM	2.01E+02	3.00E+02	3.00E+00	1.00E-05	2.91E+05	5.39E+02
2	Brazil	GRU	6.00E+01	3.00E+02	2.00E+00	1.00E-05	1.47E+05	3.83E+02
		LSTM	1.50E+02	3.00E+02	2.00E+00	1.00E-05	1.34E+09	3.66E+04
3	India	GRU	2.50E+02	3.00E+02	2.00E+00	1.00E-05	7.67E+04	2.76E+02
		LSTM	2.00E+02	3.00E+02	2.00E+00	1.00E-05	6.62E+04	2.57E+02
4	Russia	GRU	2.00E+02	3.00E+02	2.00E+00	1.00E-05	5.48E+03	7.40E+01
		LSTM	6.40E+02	3.00E+02	2.00E+00	1.00E-05	5.27E+03	7.20E+01
5	South Africa	GRU	3.50E+01	3.00E+02	2.00E+00	1.00E-05	1.71E+06	1.31E+03
		LSTM	2.00E+03	3.00E+02	2.00E+00	1.00E-05	7.64E+05	8.74E+02
6	Mexico	GRU	1.00E+02	3.00E+02	2.00E+00	1.00E-05	6.78E+05	8.23E+02
		LSTM	7.00E+02	3.00E+02	2.00E+00	1.00E-05	9.29E+04	3.04E+02
7	Peru	GRU	1.50E+03	3.00E+02	2.00E+00	1.00E-05	3.47E+07	5.89E+03
		LSTM	1.50E+03	3.00E+02	2.00E+00	1.00E-05	4.33E+07	6.58E+03
8	Chile	GRU	4.00E+02	3.00E+02	2.00E+00	1.00E-05	4.42E+03	6.64E+01
		LSTM	2.00E+03	3.00E+02	2.00E+00	1.00E-05	5.74E+04	2.40E+02
9	UK	GRU	3.00E+03	3.00E+02	2.00E+00	1.00E-05	1.95E+03	4.40E+01
		LSTM	3.00E+03	3.00E+02	2.00E+00	1.00E-05	5.81E+03	7.60E+01
10	Peru	GRU	1.75E+02	3.00E+02	2.00E+00	1.00E-05	9.43E+03	9.71E+01
		LSTM	1.43E+03	3.00E+02	2.00E+00	1.00E-05	2.76E+03	5.20E+01

Fonte: ARUNKUMAR et al. (2021).

Em outra linha de pesquisa, SHAHID; ZAMEER; MUNEEB (2020) realizaram uma proposta de comparação entre modelos de previsão estatísticas, aprendizagem de máquina e aprendizagem profunda. O algoritmo de estatística usada foi o *Autoregressive Integrated Moving Average*. O algoritmo de aprendizagem de máquina foi o *Support Vector Machine* baseado em kernels polinomiais e em *Radial Basis Function*. O algoritmo de aprendizagem profunda foram o LSTM, o GRU e o *Bidirectional Long Short Term Memory*. Todos os modelos foram usados para predição de casos confirmados, óbitos e recuperados nos dez principais países mais afetados pela COVID-19.

Em relação ao trabalho de SHAHID; ZAMEER; MUNEEB (2020) são apenas considerados as informações em relação a aprendizagem profunda LSTM e GRU, ou seja, as outras informações geradas estão sendo desconsideradas neste trabalho, pois o foco está na LSTM e GRU.

Os dez principais países afetados citados são: Brasil, China, Alemanha Índia, Israel, Itália, Rússia, Espanha, Reino Unido e Estados Unidos.

Segundo SHAHID; ZAMEER; MUNEEB (2020), para o desenvolvimento dos modelos foram utilizados a biblioteca *Keras* e o conjunto de dados usado *Basemap, W.C.-D.C.W*.

O conjunto de dados agrupa informações da data 22/01/2020 a 27/06/2020, e um total de 158 registros que foram divididos da seguinte forma: os 110 primeiros registros (22/01/2020 a 10/05 /2020) voltados para treinamento e os 48 registros restantes (11/05/2020 a 27/06/2020) voltados para treino (SHAHID; ZAMEER; MUNEEB, 2020). Com essas informações pode-se calcular que o conjunto de dados está sendo dividido em aproximadamente 70% para treinamento e 30% para teste.

Assim como o trabalho de ARUNKUMAR et al. (2021), SHAHID; ZAMEER; MUNEEB (2020) também usam o modelo *MinMaxScaler* para a normalização do conjunto de dados.

Para a construção de seus modelos, SHAHID; ZAMEER; MUNEEB (2020) levaram em consideração os hiperparâmetros: número de camadas ocultas, número de nós, taxa de aprendizagem, otimizador, tamanho do lote e épocas (Figura 18).

Da mesma forma, ARUNKUMAR et al. (2021), SHAHID; ZAMEER; MUNEEB (2020) utilizaram o mesmo otimizador, o Adam. Os dois trabalhos compartilham de uma métrica de desempenho igual, RMSE, da mesma forma descrita por ARUNKUMAR et al. (2021), SHAHID; ZAMEER; MUNEEB (2020) descreve que quanto menor o valor dessa métrica melhor será o modelo. SHAHID; ZAMEER; MUNEEB (2020) apresenta em seu trabalho mais duas métricas de desempenho, erro absoluto médio (MAE) e coeficiente de determinação (*r2_score*). Assim como o RMSE, quanto menor o valor de MAE melhor para o modelo. De acordo com SHAHID; ZAMEER; MUNEEB (2020), quanto mais próximo de um o valor do r2_*score* melhor o modelo.

Analisando os resultados descritos por SHAHID; ZAMEER; MUNEEB (2020), é possível concluir que o modelo LSTM resultou em valores menores na métrica MAE para os casos confirmados e óbitos, respectivamente 2,0463 e 0,0095. Já a métrica RMSE mostrou valores menores para os casos recuperados. Para China o valor foi 2,2428 e para o Reino Unido 0,0103. Em relação ao r2_score, o modelo LSTM apresenta o maior valor nos casos recuperados nos UK, 0,9996. Enquanto no modelo GRU notasse que houve um melhor desempenho com as métricas MAE e RMSE para China nos casos confirmados, 2,8553 e 3,3158, óbitos, 0,0321 e 0,0402 e recuperados, 7,04867 e 8,4009.

Entre os trabalhos de ARUNKUMAR et al. (2021) e SHAHID; ZAMEER; MUNEEB (2020), com base nas métricas de desempenho, foram apresentados valores menores nas métricas de SHAHID; ZAMEER; MUNEEB (2020). Dessa forma, os modelos construídos são melhores do que do ARUNKUMAR et al. (2021).

Figura 18 – Parâmetros e seus valores.

Method	Parameters	Values
SVR	C	3.0
	epsilon	0.0000001
	degree	3
	tolerance	0.000001
LSTM/Bi-LSTM/GRU	Layers	3
	No. of neurons	{16,32,64,128}
	Learning rate	0.001
	Optimizer	Adam
	Batch size	10
	Epochs	300
	Time step	3
ARIMA	(p, d, q)	(1,1,1)

Fonte:SHAHID; ZAMEER; MUNEEB (2020).

Avaliando os estudos apresentados, é possível analisar que os dois trabalhos geraram resultados positivos em relação a predição de casos, recuperados e óbitos, tanto para o modelo LSTM quanto para o modelo GRU. Além disso, os dois trabalhos apresentados compartilham de algumas características em comum, como o processo de normalização *MinMaxScaler*, o otimizador Adam, alguns dos hiperparâmetros usados e métricas avaliativas. Desta forma, é válido realizar um estudo destes algoritmos de RNR, LSTM e GRU, para um conjunto de dados diários de casos e óbitos no Centro-Oeste do Brasil.

Alguns dos termos usados neste capítulo, como, *Google COLAB*, normalização, dados de treino e dados de teste, hiperparâmetros, métricas de

desempenho, função de erro e dentre outros termos, serão explicados nos próximos capítulos.

4 AMBIENTE DE PRODUÇÃO E METODOLOGIA

Neste capítulo, será descrito o ambiente de desenvolvimento dos modelos, assim como, a explicação sobre o *Google Colab*, a linguagem de programação *Python* e as bibliotecas usadas na implementação do *notebook*. Em seguida, será apresentada a metodologia, o conjunto de dados, as etapas de pré-processamento e os hiperparâmetros usados nos modelos *Long Short Term Memory* e *Long Short Term Memory*. Por fim, serão apresentadas as métricas de desempenho usadas nos modelos.

4.1 Ambiente de desenvolvimento

Como ambiente de desenvolvimento foi usada a aplicação *web Google Colab* para a implementação e avaliação das predições de casos e óbitos diários de COVID-19 no Centro-Oeste. *Google Colab* é um produto da Empresa *Google,* que possibilita a criação de *notebooks*, ambientes de desenvolvimento, para pesquisas na área da ciência de dados. O *Google Colab* executa, principalmente, códigos desenvolvidos na linguagem de programação *Python* sendo particularmente usado para *Machine Learning* e análise de dados. É um serviço *web* de *notebooks* que não necessita de nenhuma instalação ou configuração, possui acesso gratuito e oferece recursos de computadores, como GPUs (*Graphics Processing Units*) para o processamento desses códigos (COLAB, 2021).

4.1.1 Linguagem de programação *Python*

Durante os últimos anos a linguagem de programação *Python* tem sido bastante usada em análise de dados, computação exploratória interativa e na visualização de dados. Isso porque a linguagem possui diversas bibliotecas aprimoradas para a manipulação de dados e para construção de modelos de ML

(HASIJA; CHAKRABORTY, 2021). Neste trabalho foi usada a versão 3.7.12 da Linguagem *Python*.

4.1.2 Bibliotecas

Conforme a linguagem de programação usada é a *Python*, também são usadas suas bibliotecas. Elas são: *Pandas*, *Plotly*, *Scikit-learn*, *Keras* e *Numpy*. Os tópicos a seguir mostram suas características:

- *Pandas*: é uma biblioteca voltada para manipulação e análise de dados, através de uma tabela de dados denominada *DataFrame*. Com a *Pandas,* é possível fazer importações, limpeza, manipulações de dados e processamento de dados (PANDAS, 2021). A sua utilização neste trabalho está voltada para importação e limpeza do conjunto de dados.
- *Plotly*: é uma biblioteca para criação de gráficos interativos, muito utilizada em estudos geográficos, estatísticos e financeiros e na ciência de dados. Com a *Plotly,* é possível criar gráficos a partir de um conjunto de dados, e salvar os mesmo em um repositório para visualização *web* (PLOLTY, 2021). A *Plotly* foi usada para criar gráficos interativos do conjunto de dados e dos resultados.
- *Scikit-learn*: é uma biblioteca voltada para construção de modelos de ML, sendo eles de aprendizagem supervisionada e não supervisionada. Seu uso também está voltado para o conjunto de dados, principalmente para o pré-processamento dos dados, avaliação dos modelos e seleção (SCIKIT-LEARN, 2021). A *Scikit-learn* foi usada para o pré-processamento dos dados, normalização e para a avaliação dos modelos.

- *Keras*: é uma *Application Programming Interface* (API) desenvolvida em *Python* para criação de modelos de DL e é executada na plataforma de ML *TensorFlow*. Seu objetivo é ser uma ferramenta simples, flexível é ao mesmo tempo poderosa (KERAS, 2021). Seu uso neste trabalho está voltado para a criação dos modelos Rede Neural Recorrente, LSTM e GRU.

- *NumPy*: é uma biblioteca que permite a manipulação de matrizes multidimensionais, além da criação delas. É bastante usada para executar cálculos entre suas matrizes ou vetores (NUMPY, 2021). Foi usada neste trabalho para separação das matrizes de treino e teste.

4.2 Metodologia

Neste tópico, será descrita a metodologia utilizada no trabalho. Apresentando as características do conjunto de dados, as etapas de pré-processamento, os hiperparâmetros usados e as métricas de avaliação.

De antemão, este trabalho utiliza como método, a pesquisa bibliográfica, com o uso de técnicas em análise de dados. Portanto, este trabalho trata-se de uma pesquisa exploratória (Universia, 2020).

4.2.1 Conjunto de dados

Foi escolhido o conjunto de dados de COVID-19 no Brasil, disponibilizados pelo Ministério da Saúde, no site Coronavírus Brasil (2021). Sua escolha se deu porque se trata de informações disponibilizadas pelo setor governamental responsável pelo controle e supervisão da saúde pública do país.

O conjunto contém registros diários, desde 25 de fevereiro de 2020 até os dias atuais. Foram utilizados os registros entre as datas de 25 de fevereiro de 2020 a 31 de outubro de 2021. Não foram utilizadas todas as informações disponibilizadas, pois houve a redução do número de linhas e colunas, dado que, para este trabalho, as colunas relevantes são apenas as de caso diários e óbitos diários sobre o Centro-Oeste.

As informações disponíveis no conjunto de dados podem ser contempladas na Tabela 1.

Tabela 1 – Informações disponíveis no conjunto de dados.

Coluna	Informação
regiao	Nome das regiões do Brasil.
estado	Nome dos estados brasileiros.
municipio	Nome dos municípios brasileiros.
coduf	Código das Unidades da Federação.
codmun	Código dos Municípios.
codRegiaoSaude	Código Região de Saúde.
nomeRegiaoSaude	Nome Região de Saúde.
data	Data do registro.
semanaEpi	Número da semana epidemiológico.
populacaoTCU2019	Estimativa da população de 2019 pelo Tribunal de Contas da União
casosAcumulado	Número total de casos confirmados por COVID-19, isto é, a soma de todos os casos confirmados até o momento.
casosNovos	Número de casos novos (diários) confirmados por COVID-19.
obitosAcumulado	Número total de óbitos confirmados por COVID-19, isto é, a soma de todos os óbitos confirmados até o momento.
obitosNovos	Número de óbitos novos (diários) confirmados por COVID-19.

Recuperadosnovos	Número de recuperados é o resultado de um cálculo composto, que considera os registros de casos e óbitos com confirmação de COVID-19
emAcompanhamentoNovos	São todos os casos confirmados por COVID-19, que após 14 dias não evoluíram para óbito.
interior/metropolitana	Código para identificar região do interior, ou, metropolitana (0 ou 1).

Fonte: Elaborado pelo autor.

4.2.2 Pré-processamento do conjunto de dados

O pré-processamento consistiu em quatro passos:

- O primeiro foi a limpeza do conjunto de dados, isto é, a seleção dos atributos que foram usados e a exclusão dos atributos que não foram utilizados.
- O segundo passo consistiu na divisão em dois conjuntos de dados diferentes, o dos casos confirmados diários no Centro-Oeste e o de óbitos diários no Centro-Oeste.
- No terceiro passo, houve a divisão dos dois conjuntos de dados para treinamento e teste.
- Por fim, o quarto passo consistiu na normalização dos dois conjuntos de dados.

4.2.2.1 Limpeza do conjunto de dados

Para realizar a limpeza do conjunto de dados, foi usada a biblioteca *Pandas* para manipular todo o conjunto. O primeiro passo, foi filtrar a coluna região. Dessa forma, foi possível obter como conjunto de dados, todas as informações relacionadas ao Centro-Oeste. O segundo passo, foi selecionar na coluna de municípios os nomes que não possuem valor no conjunto de dados, pois eles correspondiam à soma de todos os municípios de cada estado. No terceiro passo, foram selecionadas as colunas de data, casos novos e óbitos novos. Por fim, foi feita a soma de todos os estados para se obter os casos e óbitos diários por COVID-19 no Centro-Oeste.

Tabela 2 – Cinco primeiros registros de casos e óbitos diários por COVID-19 no Centro-Oeste.

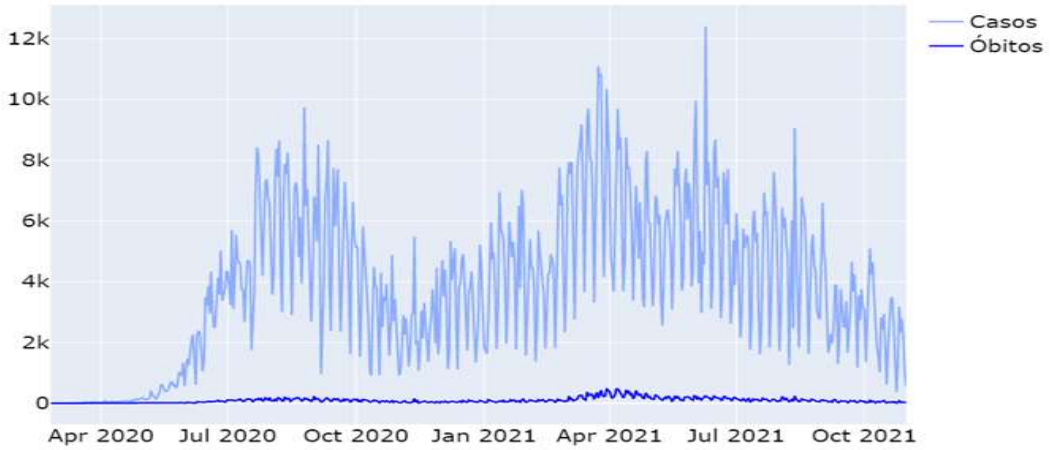
data	casosNovos	obitosNovos
2/25/2020	0	0
2/27/2020	0	0
2/28/2020	0	0
3/1/2020	0	0
3/2/2020	0	0

Fonte: Elaborado pelo autor.

A Tabela 2 apresenta os cinco primeiros registros de casos e óbitos diários por COVID-19 no Centro-Oeste apenas de forma ilustrativa e orientativa. Como o conjunto de dados possui 615 registros, ficaria inviável mostrá-la como tabela. Outra forma de visualização é através dos gráficos feitos a partir da biblioteca *Plotly* (Figura 19).

Para facilitar a manipulação e visualização dos registros é feita a sua divisão conforme descreve o próximo tópico.

Figura 19 – Casos e óbitos diários por COVID-19 no Centro-Oeste entre 25/02/2020 e 31/10/2021



Fonte: Elaborado pelo autor.

4.2.2.2 Divisão do conjunto de dados

Para facilitar na manipulação e visualização do conjunto de dados, foi feita a divisão em dois conjuntos de dados, sendo eles: casos diários de COVID-19 no Centro-Oeste (Tabela 3) e óbitos diários por COVID-19 no Centro-Oeste (Tabela 4).

Com essa divisão, fica mais prático implementar as etapas de treinamento, teste e de normalização dos dados. Outro aspecto fundamental é a melhor visualização dos gráficos gerados pelos métodos da biblioteca *Plotly* (Figura 21 e Figura 22).

Tabela 3 – Cinco primeiros registros de casos diários de COVID-19 no Centro-Oeste.

data	casosNovos
2/25/2020	0
2/27/2020	0
2/28/2020	0
3/1/2020	0

3/2/2020	0

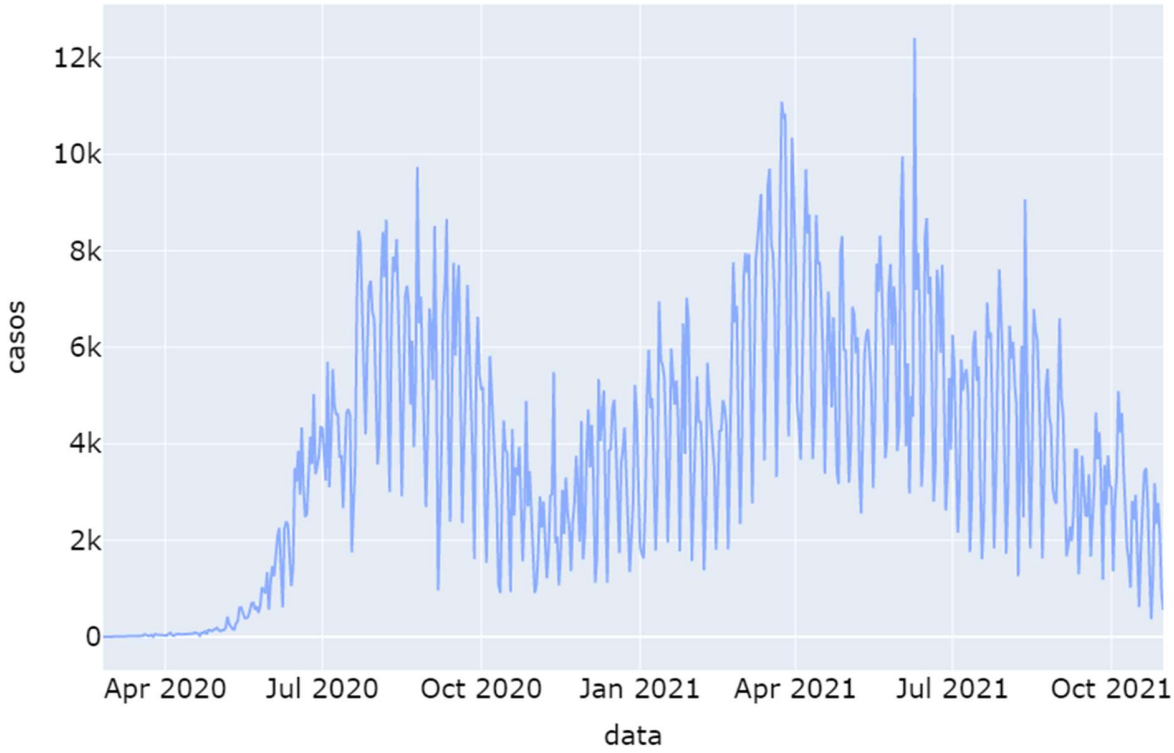
Fonte: Elaborado pelo autor.

Tabela 4 – Cinco primeiros registros de óbitos diários de COVID-19 no Centro-Oeste.

data	obitosNovos
2/25/2020	0
2/27/2020	0
2/28/2020	0
3/1/2020	0
3/2/2020	0

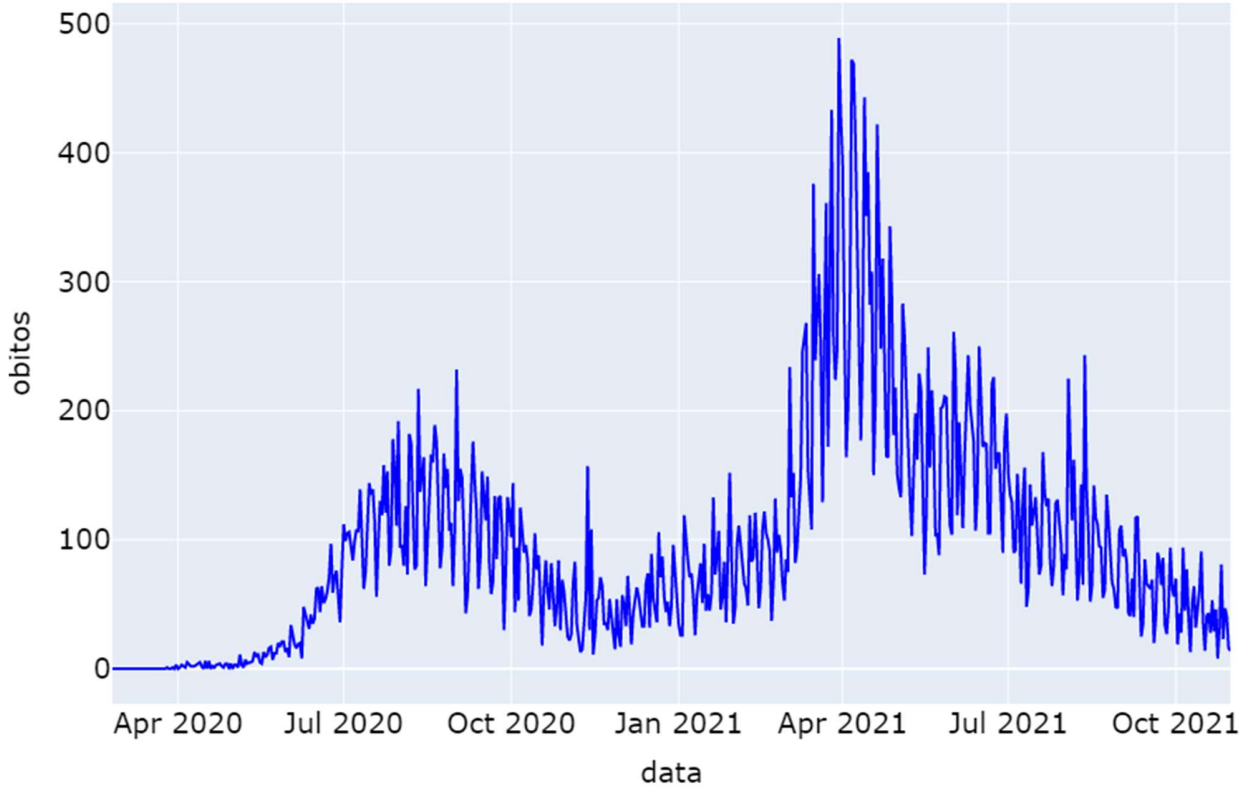
Fonte: Elaborado pelo autor.

Figura 20 – Casos diários por COVID-19 no Centro-Oeste entre 25/02/2020 e 31/10/2021



Fonte: Elaborado pelo autor.

Figura 21 – Óbitos diários por COVID-19 no Centro-Oeste
entre 25/02/2020 e 31/10/2021



Fonte: Elaborado pelo autor.

4.2.2.3 Treinamento e teste

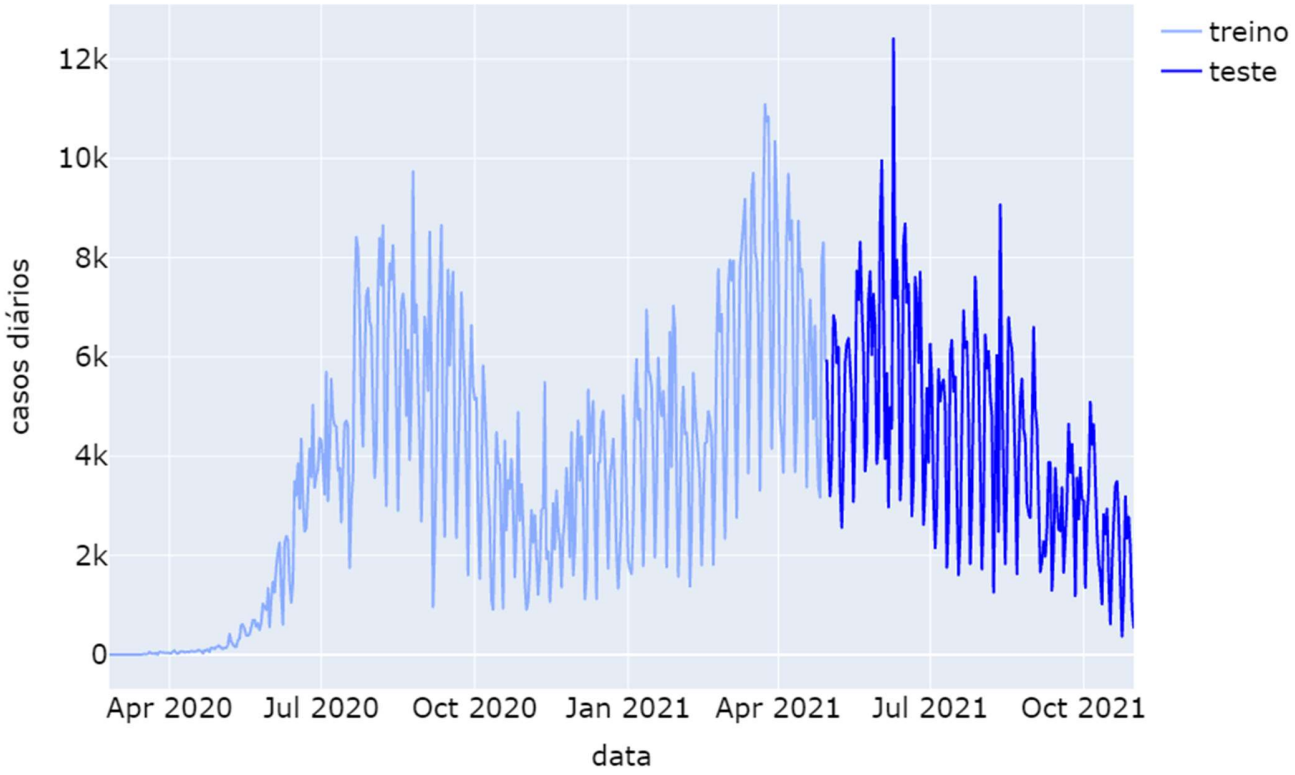
Os dois conjuntos de dados são divididos em dois subconjuntos: um para treinamento e um para teste. Segundo (SILVA, 2020) “o subconjunto para treinamento é fornecido aos modelos durante o treinamento, de modo a obter as predições. Já o subconjunto de teste é utilizado durante o teste para avaliar a performance final dos modelos.”

A divisão usada neste trabalho é a mesma usada no trabalho de SHAHID; ZAMEER; MUNEEB (2020), pois entre os dois trabalhos apresentados, foi que mostrou melhores resultados. Logo a divisão ficou em 70% para treinamento e 30%

para teste. Como se trata de um problema de série temporal, a ordem das informações apresentam grande importância. Isso significa que não podem ser feitos uma mesclagem, ou, um embaralhamento dos dados.

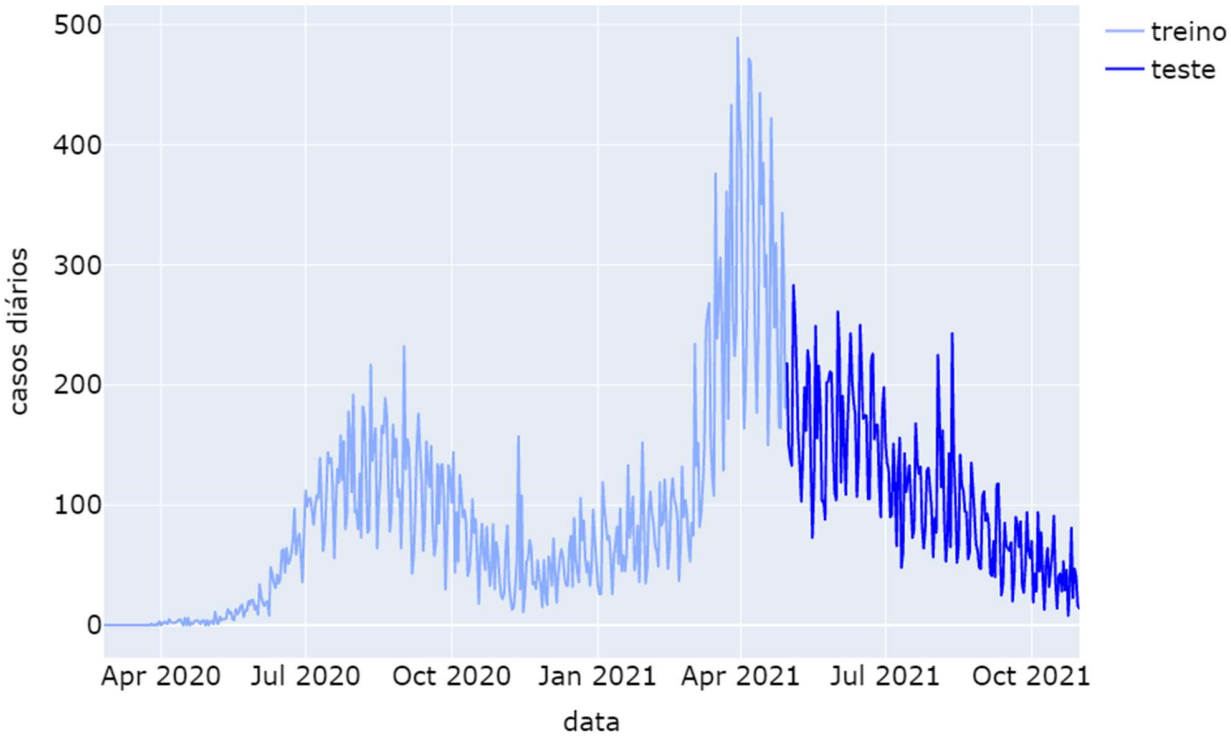
Em relação aos dois conjuntos de dados, as informações situadas entre de 25 de fevereiro de 2020 a 29 de abril de 2021 foram agrupadas para treino, e as que estão entre de 30 de abril de 2021 a 31 de outubro de 2021 foram agrupadas como dados de teste (Figura 22 e Figura 23). Numericamente são 430 dados para treino e 185 para teste.

Figura 22 – Divisão do conjunto de dados de casos diários em 70% treino e 30% teste.



Fonte: Elaborado pelo autor

Figura 23 – Divisão do conjunto de dados de óbitos diários em 70% treino e 30% teste.



Fonte: Elaborado pelo autor

4.2.2.4 Normalização dos conjuntos de dados

Para reduzir a complexidade de um problema de RNR e o tempo de treinamento do modelo, foi usada uma técnica para realizar a normalização dos seus dados, isto é, para que valores numéricos grandes tenham um intervalo de tempo menor (SILVA, 2020).

A técnica de normalização usada neste trabalho é a mesma usada nos trabalhos de ARUNKUMAR et al. (2021) e SHAHID; ZAMEER; MUNEEB (2020): a *MinMaxScaler*, cuja biblioteca é a *Scikit-learn*. Essa técnica normaliza os dados para valores entre zero e um.

4.2.3 Hiperparâmetros

Para os dois modelos, LSTM e GRU, foram usados os mesmos hiperparâmetros: número de camadas ocultas, número de neurônios por camada, otimizador, tamanho do lote e épocas.

O **número de camadas ocultas** indica quantas camadas, LSTM ou GRU, são usadas no modelo. O **número de neurônios** por camada representa o espaço de saída daquela camada. **Adam** é um algoritmo de otimização que é responsável por balancear os pesos da rede com uma função de perda. **Tamanho do lote** representa o número de amostras por atualização do gradiente. E **épocas** representa o número de interações durante o treino do modelo (Keras, 2021).

Com base nos resultados do trabalho de SHAHID; ZAMEER; MUNEEB (2020), este trabalho usa os mesmos valores dos hiperparâmetros (Tabela 5).

Tabela 5 – Hiperparâmetros dos modelos.

Hiperparâmetros	Valores
Número de camadas ocultas	3
Número de neurônios por camada	{16, 32, 64}
Otimizador	Adam
Tamanho do lote	10
Épocas	300

Fonte: Elaborado pelo autor.

4.2.4 Métricas de desempenho

Quando se trata de problemas de regressão, as métricas de desempenho são conhecidas como métricas de erro, pois elas mostram o quão proximo o valor real está do valor de predição (SILVA, 2020).

Para este trabalho foram definidas as mesma métricas de erro usadas nos trabalhos de ARUNKUMAR et al. (2021) e SHAHID; ZAMEER; MUNEEB (2020). São elas: *Mean Absolute Error, Root Mean Squared Error* e *r2_score*. As fórmulas a seguir, correspondem respectivamente a cada uma delas.

$$MAE = \frac{1}{M}\sum_{i=1}^{M}|C - \hat{C}| \tag{15}$$

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(C - \hat{C})^2} \tag{16}$$

$$r_2\_score = 1\frac{\sum|C - C'|}{\sum(C - \hat{C})} \tag{17}$$

O símbolo C equivale o valor real e o símbolo $\hat{C}$ o valor estimado, ou, o valor de predição. De acordo com ARUNKUMAR et al. (2021) e SHAHID; ZAMEER; MUNEEB (2020), quanto mais próximo de zero os valores de *Mean Absolute Error* e *Root Mean Squared Error* melhor o modelo. E segundo ARUNKUMAR et al. (2021), quanto mais próximo de um o valor do *r2_score,* melhor o modelo.

5 RESULTADOS

Os experimentos foram feitos a partir de um computador com processador Intel(R) Core (TM) i5-9400F, 16GB de memória RAM e o sistema operacional MS-Windows® 10 Pro. Com base nas etapas descritas no Capítulo 4, isto é, com o pré-processamento dos dados e os modelos definidos, foram executados os treinamentos e testes, para a obtenção da predição de casos e óbitos diários pela COVID-19 na Região Centro-Oeste do Brasil.

5.1 Resultado casos diários LSTM e GRU

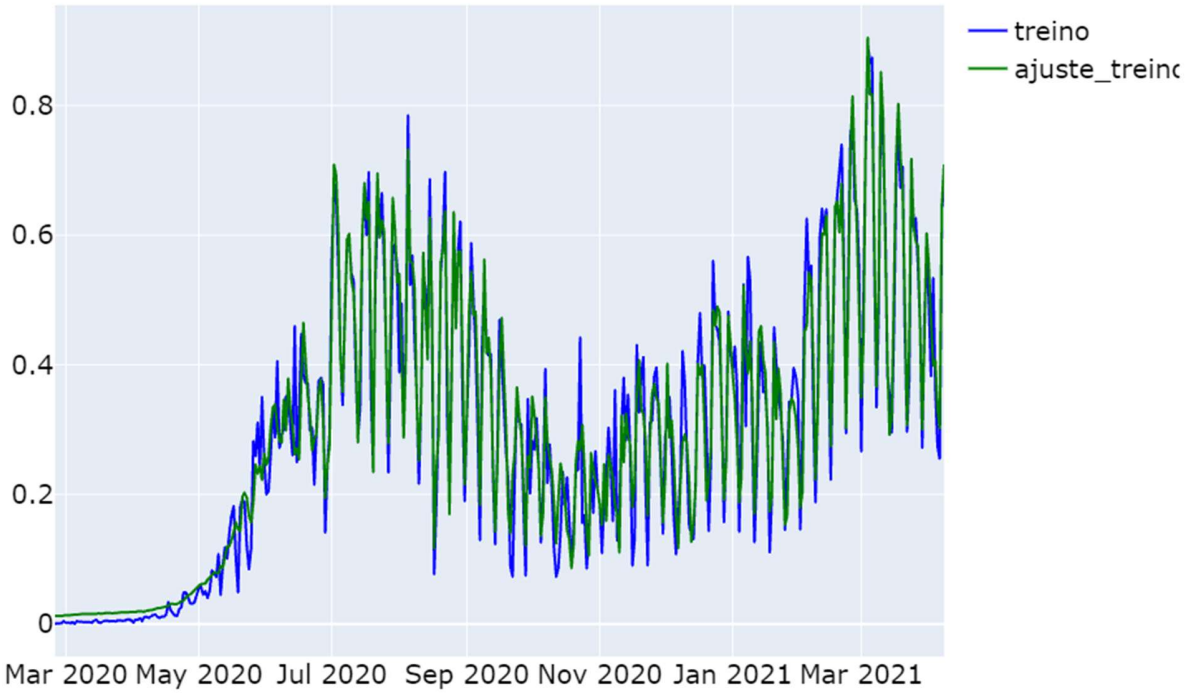
Os resultados obtidos no modelo *Long Short Term Memory* (Tabela 6) mostram que durante a fase de treino os resultados do *Mean Absolut Error* e *Root Mean Squared Error* foram menores em relação a fase de teste, tendo uma diferença de 500,5 para MAE e 640,57 para RMSE. Já r2_*score* se manteve igual nas duas situações. Teoricamente, essa diferença indica que, durante a fase de treino, o modelo conseguiu fazer um ajuste mais próximo aos dados reais (Figura 24), diferente da fase de teste (Figura 25) que teve um valor de predição mais longe dos dados reais e com isso os valores da MAE e RMSE se apresentaram superiores.

Tabela 6 – Resultados casos diários modelo LSTM.

Métrica	Treino	Teste
MAE	463,42	963,92
RMSE	620,96	1261,53
r2_*score*	0,94	0,94

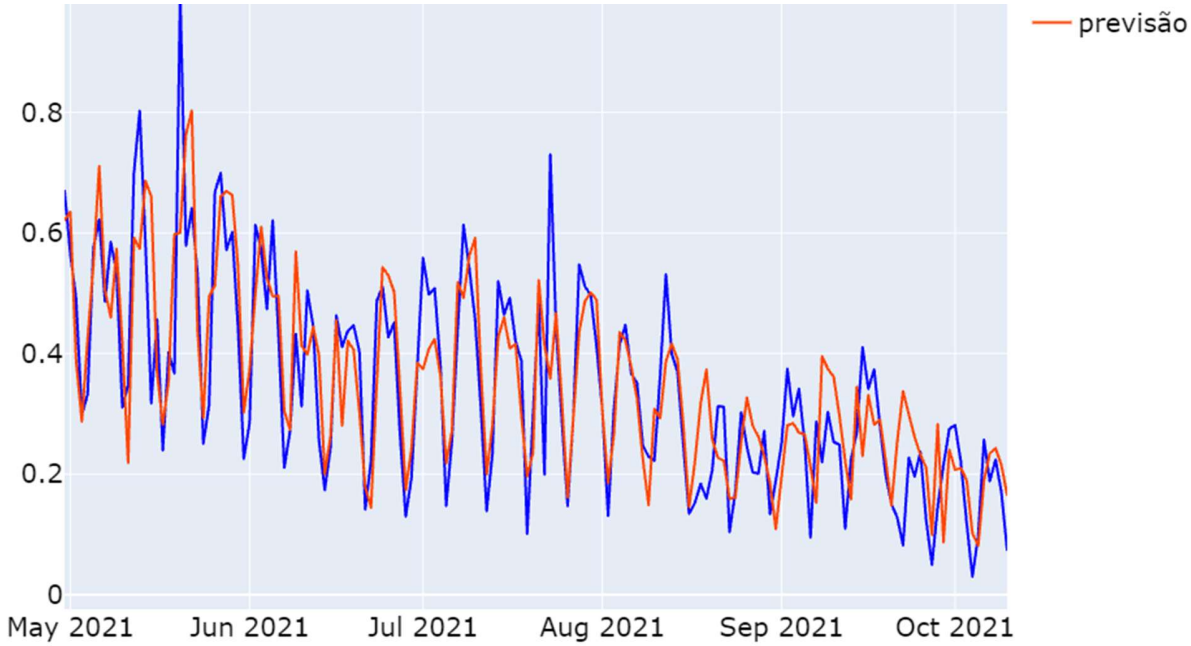
Fonte: Elaborado pelo autor.

Figura 24 – Fase de treino LSTM casos diários



Fonte: Elaborado pelo autor

Figura 25 – Fase de teste LSTM casos diários



Fonte: Elaborado pelo autor

Já os resultados do modelo *Gated Recurrent Unit* (Tabela 7) mostram que durante a fase de treino os resultados do MAE e RMSE, em comparação ao LSTM, foram menores, tendo como diferença entre eles 276,33 para MAE e 368,41 para RMSE. Isso mostra, que durante a fase de treino, o modelo GRU (Figura 26) foi superior ao modelo LSTM (Figura 24). Porém, esses valores se invertem durante a fase de teste (Figura 27), pois o modelo GRU, em comparação ao LSTM, os valores foram superiores.

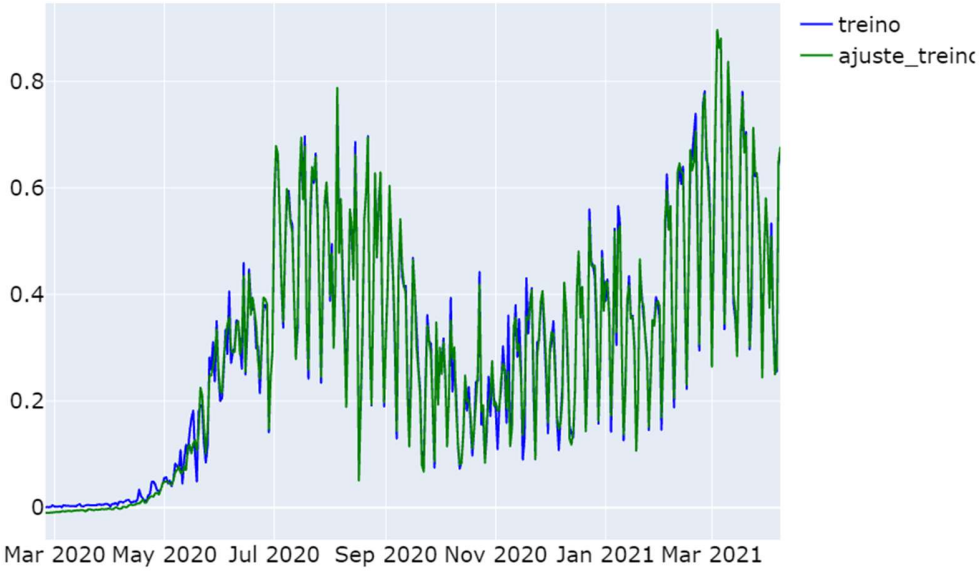
De modo geral, levando apenas em consideração os resultados do teste, o modelo LSTM se mostrou superior ao modelo GRU nas métricas MAE e RMSE, já no r2_*score*, o modelo GRU apresentou melhores resultados.

Tabela 7 – Resultados casos diários modelo GRU.

Métrica	Treino	Teste
MAE	187,09	1327,54
RMSE	252,55	1685,23
r2_score	0,98	0,97

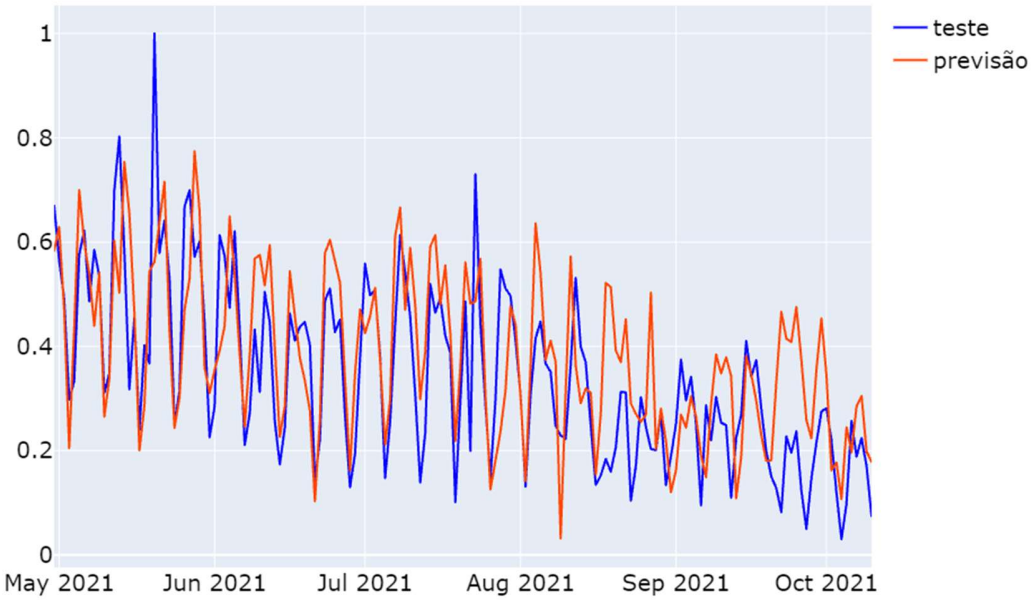
Fonte: Elaborado pelo autor

Figura 26 – Fase de teste GRU casos diários



Fonte: Elaborado pelo autor

Figura 27 – Fase de treino GRU casos diários



Fonte: Elaborado pelo autor

5.2 Resultado óbitos diários LSTM e GRU

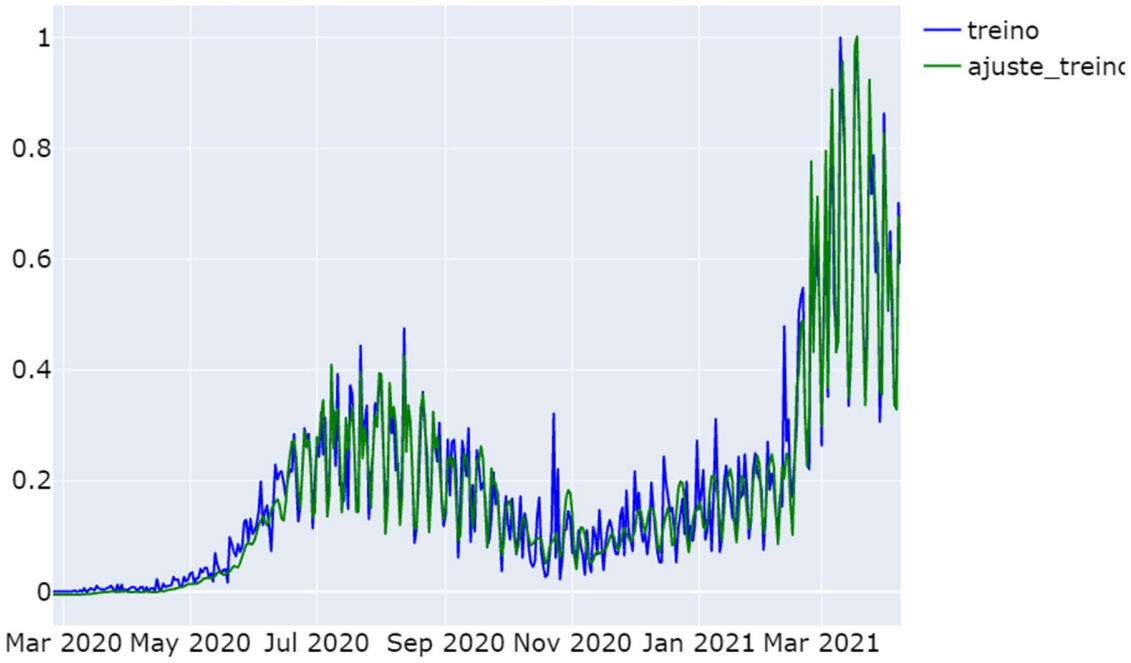
Os resultados obtidos no modelo LSTM (Tabela 8) mostram que novamente durante a fase de treino (Figura 28), o modelo apresentou bons resultados e nos testes o modelo LSTM apresentou erros consideráveis (superiores) em relação aos dados reais (Figura 29), com uma diferença de 17,52 no MAE e 23,7 no RMSE. Isso mostra que o ajuste feito durante a fase de teste (Figura 28) não foi tão apropriado para predição de óbitos diários.

Tabela 8 – Resultados óbitos diários modelo LSTM.

Métrica	Treino	Teste
MAE	15,57	33,09
RMSE	21,92	45,62
r2_score	0,94	0,94

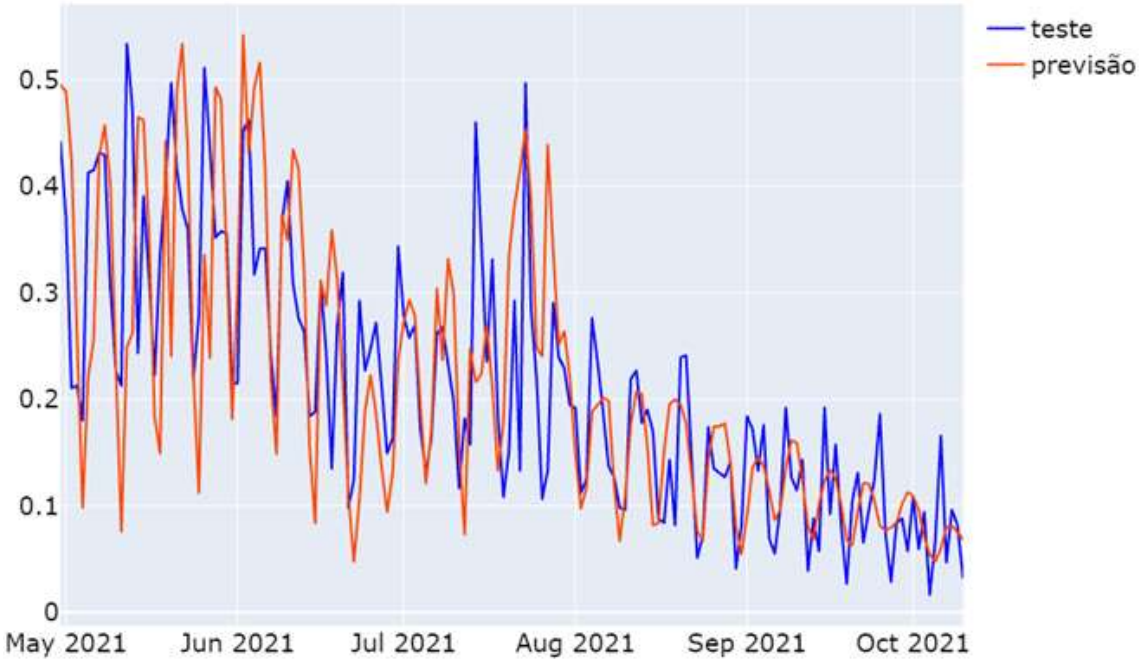
Fonte: Elaborado pelo autor.

Figura 28 – Fase de treino LSTM óbitos diários



Fonte: Elaborado pelo autor

Figura 29 – Fase de teste LSTM óbitos diários



Fonte: Elaborado pelo autor

Os resultados do modelo GRU (Tabela 9), em comparação ao modelo LSTM, tanto na fase de treino (Figura 30) quanto na fase de teste (Figura 31) mostraram resultados melhores em todas as métricas.

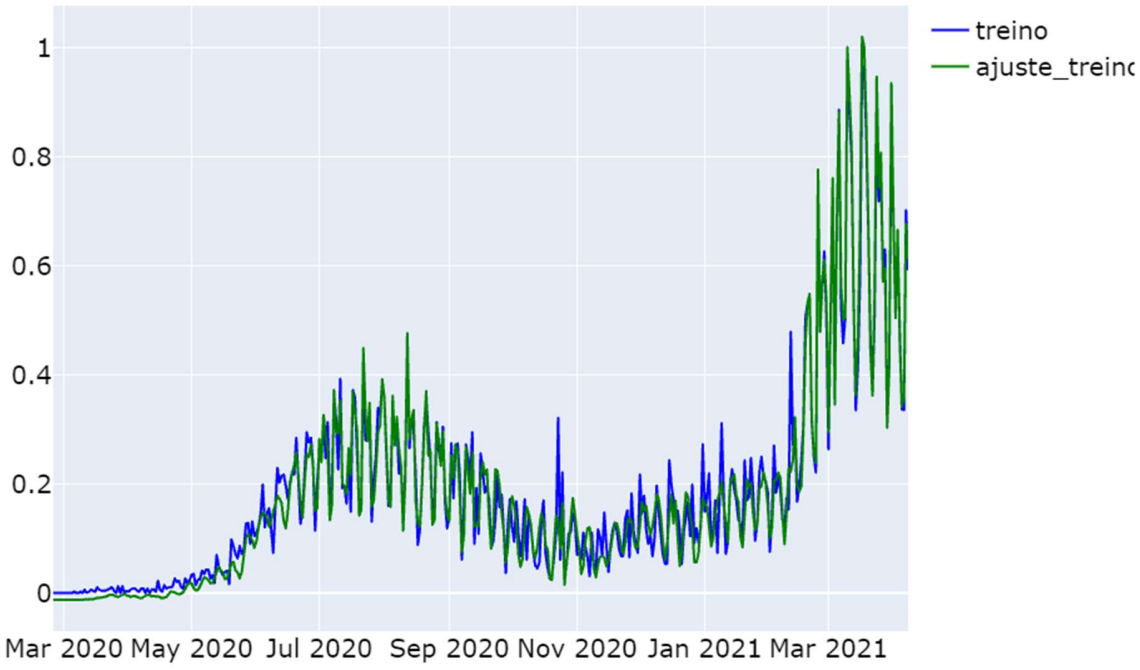
Nota-se que o modelo LSTM foi melhor no conjunto de dados (casos) onde os valores dos registros eram maiores, já o modelo GRU foi melhor no conjunto de dados (óbitos) onde os valores de registros eram menores.

Tabela 9 – Resultados óbitos diários modelo GRU.

Métrica	Treino	Teste
MAE	12,20	29,07
RMSE	17,34	40,10
r2_score	0,96	0,96

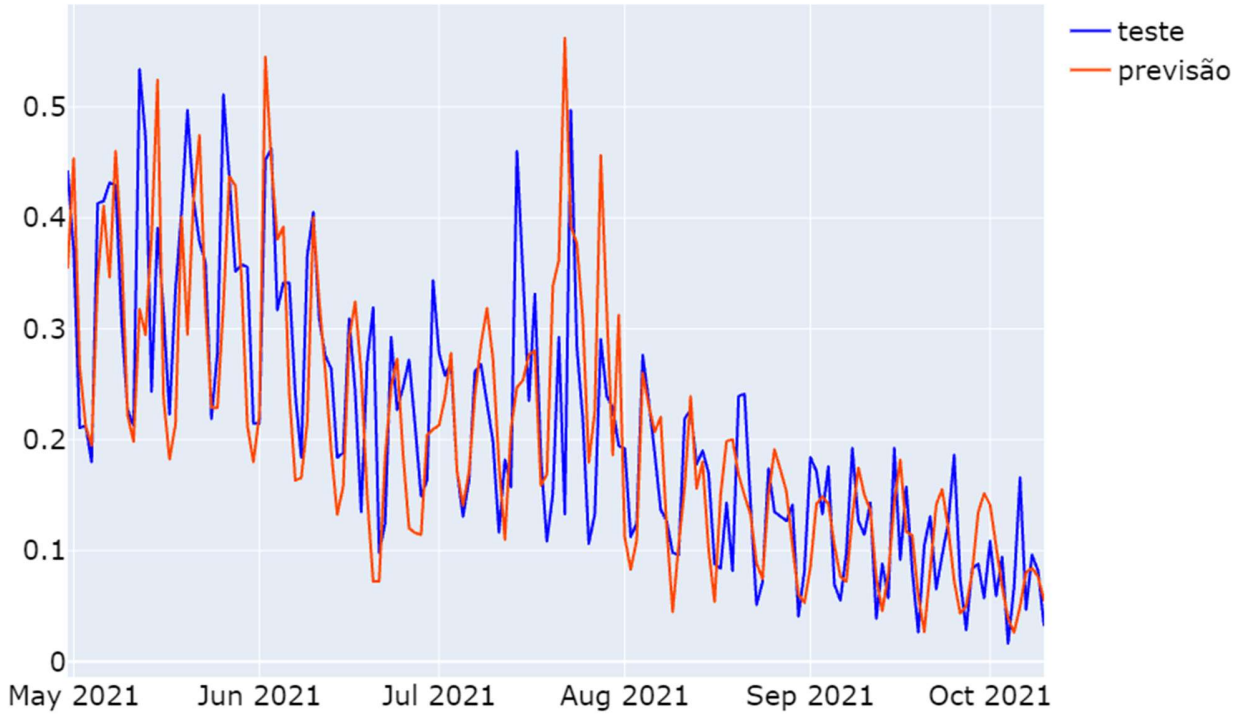
Fonte: Elaborado pelo autor.

Figura 30 – Fase de treino GRU óbitos diários



Fonte: Elaborado pelo autor

Figura 31 – Fase de teste GRU óbitos diários



Fonte: Elaborado pelo autor

6 RESULTADOS E TRABALHOS FUTUROS

Este trabalho buscou fazer um estudo de algoritmos de Redes Neurais Recorrentes, especificamente *Long Short Term Memory* e *Gated Recurrent Unit,* para predição de casos e óbitos diários pela COVID-19 no Centro-Oeste brasileiro, usando como conjunto de dados informações retiradas do site Coronavírus Brasil (2021). Os dados passaram por uma etapa de pré-processamento para que fosse possível fazer a otimização dos mesmos, e os modelos foram construídos a partir dos trabalhos relacionados. E os resultados foram avaliados por três métricas de desempenho diferentes: *Mean Absolut Error*, *Root Mean Squared Error* e r2_*score*.

Para o desenvolvimento foi usado o *Google Colab*, usando como linguagem de programação a *Python* e suas poderosas bibliotecas, principalmente a *Keras* que foi responsável pela construção dos modelos.

Com os resultados apresentados, foi possível notar que o modelo LSTM apresentou melhores resultados que o modelo GRU na análise dos casos diários da doença. Os resultados foram MAE 963,92, RMSE 1261,53 e r2_*score* 0.94. Já os resultados do GRU: MAE 1327,54, RMSE 1685.23 e r2_*score* 0,97. Entretanto, no conjunto de dados sobre óbitos diários causados pela COVID-19, o modelo GRU se mostrou melhor que o LSTM. Os resultados foram MAE 29,07, RMSE 40,10 e r2_score 0,96.

Dessa forma os dois modelos se mostraram apropriados para a predição de casos e óbitos diários pela COVID-19 no Centro-Oeste brasileiro. Porém, cada um dos modelos conseguiu fazer a predição de forma aceitável para cada um dos casos (casos diários e óbitos diários).

As métricas de desempenho, MAE e RMSE, apresentaram menores valores neste trabalho do que os trabalhos relacionados. Provavelmente, isso se deve pelo fato de que este trabalho possui um conjunto de dados maior que os utilizados nos dois trabalhos de referência.

Em resumo, os algoritmos LSTM e GRU, se mostraram adequados para o problema de predição de casos e óbitos diários pela COVID-19 no Centro-Oeste.

Sendo que o modelo LSTM foi melhor para a predição dos casos diários, e o modelo GRU foi melhor para a predição dos óbitos diários.

O código fonte dos modelos implementados encontra-se no *GitHub*: https://github.com/pedropaulo42/TCC-II .

Para trabalhos futuros, sugere-se, a realização de testes com outros algoritmos de Redes Neurais Recorrentes, que já foram descritos em outros trabalhos, como por exemplo, o *Bidirecional Long Short Term Memory*. Uma das principais sugestões é o uso de conjunto de dados que tenha mais informações (quantidade) em relação aos casos e óbitos diários pela COVID-19, porque dessa forma é possível desenvolver modelos preditivos mais poderosos e com isso, obter-se possíveis resultados com uma menor margem de erro e uma maior índice de predição.

REFERÊNCIAS

About Keras. **Keras**. Disponível em: <https://keras.io/about/>. Acesso em: 22 de nov. de 2021.

ARUNKUMAR, K. E. et al. Forecasting of COVID-19 using deep layer Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) cells. **Chaos, Solitons and Fractals**, v. 146, 2021.

Colaboratory. **Google Research**. Disponível em: <https://research.google.com/colaboratory/intl/pt-BR/faq.html>. Acesso em: 22 de nov. de 2021.

CORONAVÍRUS BRASIL. **Coronavírus Brasil**, 22 de nov. de 2021. Disponível em: <https://covid.saude.gov.br/>. Acesso em: 22 de nov. de 2021.

Conheça os tipos de metodologia de pesquisa que você pode usar no seu TCC. Universia, 2020. Disponível em: <https://www.universia.net/br/actualidad/vida-universitaria/conheca-os-tipos-metodologia-pesquisa-que-voce-pode-usar-seu-tcc-1166813.html>. Acesso em: 17 de novembro de 2021

CUCINOTTA, D. & VANELLI, M. **Who declares Covid-19 a pandemic**. Acta Biomed, v. 91, n. 91, p. 157-160, 2020.

FACELI, Katti; LORENA, Ana Carolina; GAMA, João; CARVALHO, André Carlos Ponce de Leon Ferreira de. **Inteligência Artificial**: uma abordagem de aprendizado de máquina. Rio de Janeiro: Ltc, 2011.

Getting Started. **Scikit-learn**. Disponível em: <https://scikit-learn.org/stable/getting_started.html>. Acesso em: 22 de nov. de 2021.

Getting Started with Plotly in Python. **Plotly**. Disponível em: <https://plotly.com/python/getting-started/>. Acesso em: 22 de nov. de 2021.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. Cambridge: The Mit Press, 2017.

Johns Hopkins University. **COVID-19 Map - Johns Hopkins Coronavirus Resource Center**, 22 de nov. de 2021. Disponível em: <https://coronavirus.jhu.edu/map.html>. Acesso em: 22 de nov. de 2021.

JUNIOR, J.R. F. **Redes Neurais Recorrentes** — LSTM. Medium, 2019. Disponível em <https://medium.com/@web2ajax/redes-neurais-recorrentes-lstm-b90b720dc3f6>. Acesso em: 12 novembro de 2021.

MARUMO, Fabiano Shiiti. **DEEP LEARNING PARA CLASSIFICAÇÃO DE FAKE NEWS POR SUMARIZAÇÃO DE TEXTO**. 2018. 56 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Estadual de Londrina, Londrina, 2018.

MURPHY, Kevin P.. **Machine Learning**: a probabilistic perspective. Massachusetts: The Mit Press, 2012.

MCKINNEY, Wes. **Python for Data Analysis**. O'Reilly Media, Incorporated, 2013.

Pandas documentation. **Pandas**, 17 de out. de 2021. Disponível em: <https://pandas.pydata.org/docs/>. Acesso em: 22 de nov. de 2021.

RIBEIRO, M. H. D. M. et al. Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil. **Chaos, Solitons & Fractals**, v. 135, 2020.

RUSSELL, Stuart; NORVIG, Peter. **Inteligência Artificial**. 3. ed. Rio de Janeiro: Elsevier, 2013.

VASCO, Lucas Pimenta. **Um Estudo de Redes Neurais Recorrentes no Contexto de Previsões no Mercado Financeiro**. 2020. 49 f. TCC (Graduação) - Curso de Engenheiro de Computação, Universidade Federal de São Carlos, São Carlos, 2020.

SILVA, Rafael Gomes da. **Análise Comparativa entre os Modelos CNN e LSTM para Predição de Fluxo do Tráfego Urbano na Cidade de Recife**. 2020. 61 f. TCC (Graduação) - Curso de Ciência da Computação, Escola de Ciências Exatas e da Computação, Pontifícia Universidade Católica de Goiás, Goiânia, 2020.

SILVA, Ivan Nunes da; SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. **Redes Neurais Artificiais**: para engenharia e ciências aplicadas. 2. ed. São Paulo: Artliber Editora, 2016.

SHAHID, FARAH et al. Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. **Chaos, Solitons and Fractals**, v. 140, 2020.

What is NumPy?. **NumPy**, 22 de jun. de 2021. Disponível em: <https://numpy.org/doc/stable/user/whatisnumpy.html>. Acesso em: 22 de nov. de 2021

World Health Organization. **WHO Coronavirus (COVID-19) Dashboard**, 22 de nov. de 2021. Disponível em: <https://covid19.who.int/>. Acesso em: 22 de nov. de 2021.

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 PEDRO PAULO DE SOUSA COSTA do Curso de CIÊNCIA DA COMPUTAÇÃO, matrícula 2017.1.0028.0125-0, telefone: 62981540874 e-mail pedroppaulo42@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 Estudos de algoritmos de redes neurais recorrentes para predição de casos e óbitos diários por COVID-19, no Centro-Oeste, 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, 08 de Dezembro de 2021.

Assinatura do(s) autor(es): Pedro Paulo de Sousa Costa

Nome completo do autor: Pedro Paulo de Sousa Costa

Assinatura do professor-orientador:

Nome completo do professor-orientador: Anibal Santos Jukemura