



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ESTATÍSTICA
CURSO DE GRADUAÇÃO EM BACHARELADO EM ESTATÍSTICA**

RAMON LIMA DE OLIVEIRA TAVARES

**ANÁLISE DE SÉRIES TEMPORAIS COM APRENDIZADO DE MÁQUINA PARA
DADOS DE CONTAGEM**

**CAMPINA GRANDE
2024**

RAMON LIMA DE OLIVEIRA TAVARES

**ANÁLISE DE SÉRIES TEMPORAIS COM APRENDIZADO DE MÁQUINA PARA
DADOS DE CONTAGEM**

Trabalho de Conclusão do Curso apresentado à coordenação do curso de Estatística da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Estatística.

Orientador: Prof. Dr. Ricardo Alves de Olinda

**CAMPINA GRANDE
2024**

É expressamente proibida a comercialização deste documento, tanto em versão impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que, na reprodução, figure a identificação do autor, título, instituição e ano do trabalho.

T231a Tavares, Ramon Lima de Oliveira.

Análise de séries temporais com aprendizado de máquina para dados de contagem [manuscrito] / Ramon Lima de Oliveira Tavares. - 2024.

38 f. : il. color.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Estatística) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2024.

"Orientação : Prof. Dr. Ricardo Alves de Olinda, Departamento de Estatística - CCT".

1. Previsão de séries temporais. 2. Redes neurais recorrentes. 3. Aprendizado profundo. I. Título

21. ed. CDD 519.55

RAMON LIMA DE OLIVEIRA TAVARES

ANÁLISE DE SÉRIES TEMPORAIS COM APRENDIZADO DE MÁQUINA PARA DADOS DE
CONTAGEM

Trabalho de Conclusão do Curso apresentado à coordenação do curso de Estatística da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Estatística.

Aprovada em: 08/11/2024.

BANCA EXAMINADORA



Prof. Dr. Ricardo Alves de Olinda
Orientador



Prof. Dr. Silvio Fernando Alves Xavier Junior
Examinador interno (CCT/UEPB)



Prof. Dr. Oseas Machado Gomes
Examinador interno (CCT/UEPB)

AGRADECIMENTOS

Primeiramente, gostaria de expressar minha profunda gratidão a Deus, por me guiar e dar forças para estudar e aprender ao longo de toda essa jornada.

Os agradecimentos são à minha família, que sempre esteve ao meu lado, em especial à minha mãe e ao meu avô José Tavares, por todo o incentivo e apoio nos meus estudos. Agradeço ao meu orientador, Prof. Dr. Ricardo Alves de Olinda pela paciência, dedicação e pelas valiosas orientações que foram essenciais para o desenvolvimento deste trabalho. Sem o amor e suporte de vocês, eu não teria chegado até aqui.

Também sou grato aos membros da banca, Prof. Dr. Oseas Machado Gomes e Prof. Dr. Silvio Fernando Alves Xavier Junior, pelas contribuições e considerações enriquecedoras.

Minhas palavras de agradecimento se estendem aos meus professores e colegas do curso de Bacharelado em Estatística, pelas trocas de experiências, apoio e aprendizado. Cada um de vocês desempenhou um papel significativo na minha formação acadêmica e pessoal.

RESUMO

Este estudo apresenta metodologias para modelagem e previsão da série temporal de focos ativos detectados pelo satélite *AQUA_M-T* no Bioma do Pantanal, Brasil. A abordagem utiliza um modelo misto de Redes Neurais Recorrentes (RNN), combinando as arquiteturas *Long Short-Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU) para prever os acumulados mensais de focos ativos detectados diariamente. O objetivo principal foi verificar se essas previsões capturam a sazonalidade inerente dos dados, aplicando técnicas de aprendizado de máquina especificamente adaptadas para dados de contagem. Ao manter a série em sua estrutura original baseada em contagem, o modelo busca preservar a variabilidade natural do conjunto de dados, essencial para desenvolver modelos com alta capacidade preditiva. A análise dos dados revelou uma sazonalidade consistente ao longo do tempo, com picos e mínimas anuais recorrentes em intervalos semelhantes a cada ano. A metodologia envolveu uma preparação cuidadosa dos dados, configuração do modelo e treinamento com validação cruzada utilizando três sementes, assegurando uma generalização robusta tanto para os conjuntos de teste quanto de validação. Os resultados mostraram que o modelo combinado de LSTM e GRU apresentou bom desempenho, capturando padrões temporais complexos e demonstrando sua adequação para a modelagem de séries temporais baseadas em dados de contagem. Este trabalho contribui significativamente para a aplicação de técnicas de aprendizado profundo no monitoramento ambiental, especificamente na previsão de focos ativos. A abordagem proposta também destaca o potencial de adaptação para outros desafios de previsão em séries temporais, oferecendo novas oportunidades para pesquisa e desenvolvimento em aprendizado de máquina e previsão de séries temporais.

Palavras-chave: previsão de séries temporais; redes neurais recorrentes; aprendizado profundo.

ABSTRACT

This study presents methodologies for modeling and forecasting the time series of active hotspots detected by the satellite *AQUA_M-T* in the Pantanal Biome, Brazil. The approach utilizes a mixed model of Recurrent Neural Networks (RNN), combining the architectures of *Long Short-Term Memory* (LSTM) and *Gated Recurrent Unit* (GRU) to predict the monthly accumulations of daily detected active hotspots. The main objective was to verify whether these forecasts capture the inherent seasonality of the data, applying machine learning techniques specifically adapted for count data. By maintaining the series in its original count-based structure, the model aims to preserve the natural variability of the dataset, which is essential for developing models with high predictive capacity. Data analysis revealed consistent seasonality over time, with recurring annual peaks and minima at similar intervals each year. The methodology involved careful data preparation, model configuration, and training with cross-validation using three seeds, ensuring robust generalization for both test and validation sets. The results showed that the combined LSTM and GRU model performed well, capturing complex temporal patterns and demonstrating its suitability for modeling time series based on count data. This research significantly contributes to the application of deep learning techniques in environmental monitoring, specifically in the forecasting of active hotspots. The proposed approach also highlights the potential for adaptation to other forecasting challenges in time series, offering new opportunities for research and development in machine learning and time series forecasting.

Keywords: time series forecasting; recurrent neural networks; deep learning.

LISTA DE ILUSTRAÇÕES

Figura 1 – Esquema detalhado da unidade de Rede Neural Recorrente Simples (RNN).	12
Figura 2 – Esquema detalhado da camada LSTM.	14
Figura 3 – Esquema detalhado da camada GRU.	15
Figura 4 – Ilustração da transformação linear e da ativação linear em uma camada densa da rede neural.	16
Figura 5 – Análise de sazonalidade, tendência e resíduos dos focos de calor no Pantanal (1998 - 2024).	19
Figura 6 – Esquema detalhado da Rede Neural Recorrente híbrida (LSTM+GRU) com camadas densas.	22
Figura 7 – Previsão de focos de calor: funcionamento do predict em redes neurais recorrentes.	23
Figura 8 – Distribuição anual: valores mínimos, máximos e média dos focos de calor no Pantanal (1998 - 2024).	25
Figura 9 – Anomalias de temperatura e precipitação observadas no Brasil, com destaque para a região do Pantanal.	26
Figura 10 – Valores Mínimos e Máximos de focos de calor no Pantanal (1998 - 2024)	26
Figura 11 – Comparação das métricas de desempenho dos Modelos.	29
Figura 12 – Conjunto de treino.	29
Figura 13 – Conjunto de validação.	30
Figura 14 – Conjunto de teste.	30
Figura 15 – Histórico de perda (<i>Loss</i>) durante o treinamento e validação.	31
Figura 16 – Métricas de Desempenho MAE Durante o Treinamento	32
Figura 17 – Histórico de perda (<i>Loss</i>) durante o treinamento.	33
Figura 18 – Previsões do modelo híbrido (LSTM+GRU) - Conjunto Completo.	33
Figura 19 – Previsões dos Meses Futuros (Outubro de 2024 a Setembro de 2025).	34
Figura 20 – Previsões isoladas do modelo híbrido (LSTM+GRU).	34

LISTA DE TABELAS

Tabela 1 – Estatísticas descritivas da série temporal anual dos focos de calor no Pantanal.	24
Tabela 2 – Desempenho dos modelos em termos do erro absoluto médio (MAE) em dados de treinamento e teste para sementes de 2030, 2040 e 2050.	28
Tabela 3 – Valores reais e preditos no conjunto de teste do modelo híbrido considerando o erro absoluto médio (MAE) correspondente à semente 2030.	28
Tabela 4 – Desempenho do modelo híbrido treinado com todos os dados disponíveis.	32

SUMÁRIO

1	INTRODUÇÃO	10
2	OBJETIVOS	11
2.1	Objetivos Específicos	11
3	REFERENCIAL TEÓRICO	12
3.1	RNN	12
3.2	LSTM	12
3.3	GRU	14
3.4	Entendendo o Funcionamento das Camadas Densas em Redes Neurais Recorrentes	15
3.5	Funções de Ativação	15
3.6	Algoritmo de Otimização <i>Adam</i> : Uma Visão Geral	16
3.6.1	<i>Descrição dos parâmetros</i>	17
3.7	Regularização de Redes Neurais Recorrentes com <i>Dropout</i>	18
3.8	Análise de Sazonalidade dos Focos de Calor no Pantanal	18
3.8.1	<i>Componentes de uma Série Temporal</i>	18
3.8.2	<i>Cálculo da Tendência, Sazonalidade e Resíduo</i>	19
4	MATERIAL E MÉTODOS	20
4.1	Preparação dos Dados	20
4.2	Configuração dos Modelos	21
4.3	Treinamento e Avaliação dos Modelos	22
4.4	Como as Previsões são Calculadas?	23
5	RESULTADOS E DISCUSSÕES	24
5.1	Análise Descritiva dos focos de calor no Pantanal	24
5.1.1	<i>Análise dos Valores Extremo dos focos de calor no Pantanal</i>	25
5.1.2	<i>Preservação da Estrutura Sazonal em Dados de Contagem na Modelagem dos focos de calor no Pantanal</i>	27
5.2	Análise das Métricas de Desempenho dos Modelos LSTM e GRU e Abordagem Híbrida que Combina Ambos (LSTM+GRU)	27
5.3	Análise de Desempenho dos Conjuntos de Treino, Validação e Teste	29
5.3.1	<i>Conjunto de Treino</i>	29
5.3.2	<i>Conjunto de Validação</i>	30
5.3.3	<i>Conjunto de Teste</i>	30
5.3.4	<i>Histórico de Perda</i>	31
5.4	Resultados das Previsões Considerando Todos os Dados Disponíveis na Série Histórica e Previsões com o Modelo Híbrido (LSTM+GRU)	31
5.4.1	<i>Métricas de Desempenho MAE Durante o Treinamento</i>	32
5.4.2	<i>Histórico de Perda Durante o Treinamento</i>	33
5.4.3	<i>Modelo Híbrido Treinado com Todos os Dados da Série Temporal</i>	33
5.4.4	<i>Previsões dos Meses Futuros</i>	34
6	CONCLUSÃO	35

REFERÊNCIAS 36

1 INTRODUÇÃO

A modelagem em séries temporais é amplamente utilizada em diversas áreas, como economia, climatologia, finanças e monitoramento ambiental, com importantes referências na literatura, como Box *et al.* (2015), James *et al.* (2013) e Brockwell e Davis (2002). De maneira geral, uma série temporal é uma sequência de observações de uma variável ao longo do tempo, ou seja, uma sequência de pontos em ordem sucessiva, geralmente em intervalos uniformes.

Neste trabalho, estudamos a série temporal dos focos de calor detectados pelo satélite *AQUA_M-T* no Bioma do Pantanal, Brasil. Os focos de calor são identificados com base em anomalias de temperatura em *pixels* observados pelo satélite. Quando a temperatura de um pixel (representando uma área de 1 km²) atinge níveis elevados, como acima de 47° — um valor que, segundo o Sistema Estadual de Informações Ambientais e Recursos Hídricos Seia (2024), caracteriza um foco de calor — o satélite registra a ocorrência de um foco ativo. Esses dados, disponibilizados mensalmente pelo Instituto Nacional de Pesquisas Espaciais (INPE), oferecem uma visão histórica importante. Neste sentido, utilizamos modelos de Redes Neurais Recorrentes (RNNs), especificamente as arquiteturas *Long Short-Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU), desenvolvidas por Hochreiter e Schmidhuber (1997) e Chung *et al.* (2014) respectivamente, para modelar e prever a quantidade de focos de calor no Pantanal. A LSTM é reconhecida por sua capacidade de lidar com problemas de retenção de longo prazo, enquanto a GRU simplifica a estrutura da LSTM, aumentando a eficiência do modelo.

A combinação dessas duas arquiteturas em um modelo híbrido proporciona a robustez necessária para capturar os padrões complexos presentes na série temporal analisada. Um aspecto positivo das redes neurais é a capacidade de um modelo bem treinado aprender padrões independentemente da escala dos dados e do tamanho da série, como descrito por Benidis *et al.* (2022) e também nos resultados do estudo de caso de focos de calor na Amazônia de Tavares e Olinda (2024), no qual foi observado um treinamento semelhante com dados de focos de calor na Amazônia, que varia de um mínimo de 70 focos registrados em abril de 1999 a um máximo de 73.141 focos em setembro de 2007, também ajustados em sua escala original.

No contexto desse estudo dos focos de calor no Pantanal, a série temporal varia de um mínimo de 1 foco registrado em abril de 2000 a um máximo de 8106 focos em setembro de 2020. Esse intervalo expressivo ressalta a importância de desenvolver uma arquitetura robusta e bem configurada, garantindo que o modelo consiga aprender essas variações e realizar previsões com menores erros em comparação aos valores reais observados. Além de explorar a aplicação das RNNs, LSTM e GRU, este trabalho apresenta como o modelo foi estruturado e treinado, incluindo o número de neurônios e épocas de treinamento utilizados, e como as previsões foram realizadas. Ademais, a fundamentação teórica por trás das redes neurais recorrentes é discutida, ajustando-se os modelos aos dados históricos para identificar a sazonalidade dos focos de calor no Pantanal.

2 OBJETIVOS

Modelar a série temporal de dados de contagem dos focos de calor no Bioma Pantanal, utilizando arquiteturas de aprendizado de máquina para compreender padrões e prever tendências.

2.1 Objetivos Específicos

1. Analisar e identificar padrões sazonais e tendências.
2. Treinar redes neurais recorrentes, com arquiteturas LSTM e GRU.
3. Gerar previsões dos focos de calor, a partir do modelo que apresentar menor erro absoluto médio (MAE).

3 REFERENCIAL TEÓRICO

De acordo com (Graves; Mohamed; Hinton, 2013), as Redes Neurais Recorrentes *Recurrent Neural Networks* (RNNs), são modelos eficientes para dados sequenciais, tendo em vista que, são capazes de lidar com problemas de rotulagem de seqüências onde o alinhamento entre entrada e saída é desconhecido. Esses modelos são construídos para aprender dependências temporais em dados sequenciais e mantêm uma memória interna para processar informações anteriores.

3.1 RNN

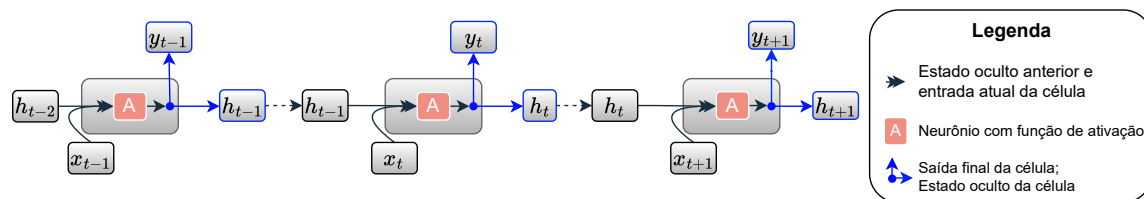
Conforme descrito por Graves, Mohamed e Hinton (2013), dada uma seqüência de entrada $(x_t, x_{t+1}, x_{t+2}, \dots, x_{t+n})$, uma RNN padrão computa a seqüência de vetores ocultos $(h_{t-1}, h_t, h_{t+1}, h_{t+2}, \dots, h_{t+n})$ e a seqüência de vetores de saída $(y_t, y_{t+1}, y_{t+2}, \dots, y_{t+n})$.

$$\mathbf{h}_t = \mathbf{A}(\mathbf{W}_{hh} \odot \mathbf{h}_{t-1} + \mathbf{W}_{xh} \odot \mathbf{x}_t + \mathbf{b}_h),$$

$$\mathbf{y}_t = (\mathbf{W}_{yi} \odot \mathbf{h}_t + \mathbf{b}_y), \quad (3.1)$$

em que \mathbf{W} é a matriz de pesos e \mathbf{b} é o viés, o operador \odot representa a multiplicação elemento a elemento, o estado de saída \mathbf{y}_t é gerado no tempo t e determinado pela informação de entrada \mathbf{x}_t e pelo estado oculto anterior \mathbf{h}_{t-1} no tempo $(t - 1)$. A Equação (3.1) demonstra como o estado oculto atual \mathbf{h}_t é calculado usando uma função de ativação \mathbf{A} , pesos \mathbf{W} e viés \mathbf{b} correspondentes. Esse modelo de unidade de Redes Neurais Recorrentes é fundamental para compreender a propagação de informações ao longo do tempo em uma Rede Neural Recorrente. A estrutura interna da unidade RNN é exibida na Figura 1.

Figura 1 – Esquema detalhado da unidade de Rede Neural Recorrente Simples (RNN).



Fonte: Elaborada pelo autor, 2024, adaptado de (Greff *et al.*, 2017)

3.2 LSTM

De acordo com o que os autores Greff *et al.* (2017), é possível entender claramente que a arquitetura da Rede de Memória de Longo e Curto Prazo (*Long Short-Term Memory*, LSTM) é particularmente eficaz para processos que requerem o processamento de seqüências temporais de longo prazo. Conforme Greff *et al.* (2017), a LSTM destaca-se pela capacidade de superar as limitações das Redes Neurais Recorrentes (RNNs) tradicionais, permitindo que informações relevantes sejam retidas por períodos mais prolongados. Isso é primordial para lidar com dependências temporais extensas. Enquanto as RNNs funcionam em seqüências

temporais mantendo uma memória interna, as LSTMs aprimoram essa capacidade ao utilizar *gates* (portões), uma vez que, controla o fluxo de informações. Neste sentido, tais portões facilitam uma retenção mais eficaz das informações a longo prazo, comparado às RNNs tradicionais, tendo em vista que enfrentam dificuldades em manter dependências temporais mais longas. Dessa forma, as LSTMs demonstram uma capacidade superior de generalização e previsão quando confrontadas com dados de entrada que se estendem por longos períodos de tempo.

A arquitetura *Long Short-Term Memory* (LSTM), conforme descrito por Greff *et al.* (2017), é projetada para lidar com as limitações das Redes Neurais Recorrentes tradicionais em tarefas de aprendizado de sequências temporais. O bloco LSTM é composto por três componentes principais, como ilustrado na Figura 2:

- **Input Layer:** Este portão regula a quantidade de nova informação que será incorporada na célula de memória. Ele determina quais informações devem ser adicionadas ao estado da célula.
- **Hidden Layer:** Este portão decide quais informações presentes na célula de memória devem ser descartadas. Ele ajuda a manter a relevância dos dados ao longo do tempo, removendo informações que não são mais necessárias.
- **Output Layer:** Este portão controla a quantidade de informação da célula de memória que será utilizada na saída do bloco LSTM. Ele decide quais informações da célula de memória serão passadas para a próxima etapa na sequência.

Nesse sentido, os portões são responsáveis por regular o fluxo de informações dentro do bloco LSTM, permitindo assim, a retenção e atualização eficaz de dados relevantes ao longo prazo. A estrutura interna do LSTM permite que o modelo capture dependências temporais extensas e mantenha a precisão em tarefas que envolvem sequências longas e complexas. Seja \mathbf{x}_t o vetor de entrada no tempo t , N o número de neurônios da camada LSTM e M o número de unidades que se encontra no vetor \mathbf{x}_t , de tal maneira que $(\mathbb{R}^{N \times M})$ representa a dimensão da matriz de pesos \mathbf{W} , $(\mathbb{R}^{N \times N})$ representa a dimensão da matriz de pesos \mathbf{R} (que são os pesos do estado oculto), e (\mathbb{R}^N) representa a dimensão da matriz do *bias* (viés). Então, tem-se os seguintes pesos para uma camada LSTM:

- Pesos de entrada: $\mathbf{W}_z, \mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o \in \mathbb{R}^{N \times M}$;
- Pesos recorrentes: $\mathbf{R}_z, \mathbf{R}_i, \mathbf{R}_f, \mathbf{R}_o \in \mathbb{R}^{N \times N}$;
- Pesos de viés: $\mathbf{b}_z, \mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o \in \mathbb{R}^N$.

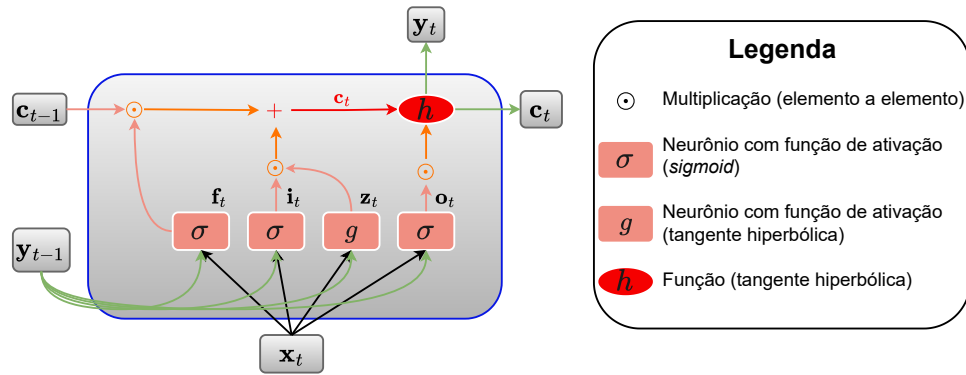
Então, de acordo com Greff *et al.* (2017), as fórmulas vetoriais para uma passagem direta em uma camada LSTM podem ser escritas como:

$$\begin{aligned}\bar{\mathbf{z}}_t &= \mathbf{W}_z \mathbf{x}_t + \mathbf{R}_z \mathbf{y}_{t-1} + \mathbf{b}_z, \\ \mathbf{z}_t &= g(\bar{\mathbf{z}}_t) \quad \text{entrada do bloco;} \\ \bar{\mathbf{i}}_t &= \mathbf{W}_i \mathbf{x}_t + \mathbf{R}_i \mathbf{y}_{t-1} + \mathbf{b}_i, \\ \mathbf{i}_t &= \sigma(\bar{\mathbf{i}}_t) \quad \text{porta de entrada;} \\ \bar{\mathbf{f}}_t &= \mathbf{W}_f \mathbf{x}_t + \mathbf{R}_f \mathbf{y}_{t-1} + \mathbf{b}_f,\end{aligned}$$

$$\begin{aligned} \mathbf{f}_t &= \sigma(\tilde{\mathbf{f}}_t) \quad \text{porta de esquecimento;} \\ \mathbf{c}_t &= \mathbf{z}_t \odot \mathbf{i}_t + \mathbf{c}_{t-1} \odot \mathbf{f}_t \quad \text{célula;} \\ \tilde{\mathbf{o}}_t &= \mathbf{W}_o \mathbf{x}_t + \mathbf{R}_o \mathbf{y}_{t-1} + \mathbf{b}_o, \\ \mathbf{o}_t &= \sigma(\tilde{\mathbf{o}}_t) \quad \text{porta de saída;} \\ \mathbf{y}_t &= h(\mathbf{c}_t) \odot \mathbf{o}_t \quad \text{saída do bloco.} \end{aligned}$$

Em que σ , g e h são funções de ativação não lineares. A função sigmoide ($\sigma(x) = \frac{1}{1+e^{-x}}$) é usada como função de ativação da porta, e a tangente hiperbólica ($g(x) = h(x) = \tanh(x)$) é comumente usada como função de ativação de entrada e saída do bloco. A multiplicação ponto a ponto de dois vetores é denotada por \odot .

Figura 2 – Esquema detalhado da camada LSTM.



Fonte: Elaborada pelo autor, 2024, adaptado de (Greff *et al.*, 2017).

3.3 GRU

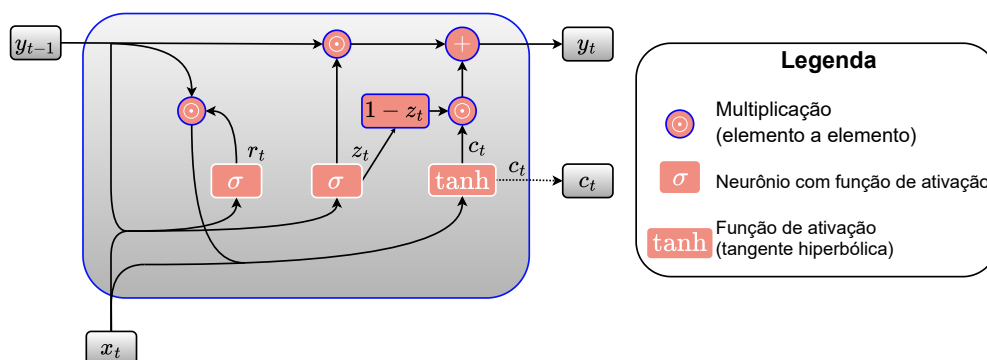
De acordo com Chung *et al.* (2014), *Gated Recurrent Units* (GRU), é uma variação da LSTM. Enquanto as LSTM possuem três portões e uma célula de memória, as GRU simplificam essa estrutura ao fundir os portões de entrada e esquecimento em um único portão de atualização. Essa simplificação tem como objetivo tornar o treinamento mais eficiente e reduzir o número de parâmetros, mantendo um desempenho comparável às LSTM. As fórmulas vetoriais para uma passagem direta em uma camada GRU, apresentadas na subseção 3.1 da Seção 3, foram extraídas do artigo "*Dynamic Gated Recurrent Neural Network for Compute-efficient Speech Enhancement*" de (Cheng *et al.*, 2024). Essas fórmulas são definidas da seguinte forma:

$$\begin{aligned} \tilde{\mathbf{z}}_t &= \mathbf{W}_{i\tilde{z}} \mathbf{x}_t + \mathbf{b}_{i\tilde{z}} + \mathbf{R}_{y\tilde{z}} \mathbf{y}_{t-1} + \mathbf{b}_{y\tilde{z}}, \\ \mathbf{z}_t &= \sigma(\tilde{\mathbf{z}}_t) \quad \text{portão de atualização;} \\ \tilde{\mathbf{r}}_t &= \mathbf{W}_{i\tilde{r}} \mathbf{x}_t + \mathbf{b}_{i\tilde{r}} + \mathbf{R}_{y\tilde{r}} \mathbf{y}_{t-1} + \mathbf{b}_{y\tilde{r}}, \\ \mathbf{r}_t &= \sigma(\tilde{\mathbf{r}}_t) \quad \text{portão de reset;} \\ \tilde{\mathbf{c}}_t &= \mathbf{W}_{i\tilde{c}} \mathbf{x}_t + \mathbf{b}_{i\tilde{c}} + \mathbf{R}_{y\tilde{c}} (\mathbf{r}_t \odot \mathbf{y}_{t-1}) + \mathbf{b}_{y\tilde{c}}, \\ \mathbf{c}_t &= \tanh(\tilde{\mathbf{c}}_t) \quad \text{estado oculto,} \end{aligned}$$

$$\mathbf{y}_t = \mathbf{z}_t \odot \mathbf{y}_{t-1} + (1 - \mathbf{z}_t) \odot \mathbf{c}_t.$$

Aqui \mathbf{W} e \mathbf{R} são matrizes de pesos; \mathbf{b} são vetores de viés; σ é a função de ativação sigmoide e \odot denota a multiplicação ponto a ponto. A Figura 3 apresenta o esquema da *Gated Recurrent Units* (GRU) e a arquitetura típica dessa rede.

Figura 3 – Esquema detalhado da camada GRU.



Fonte: Elaborada pelo autor, 2024, adaptado de (Cheng *et al.*, 2024).

3.4 Entendendo o Funcionamento das Camadas Densas em Redes Neurais Recorrentes

Em conformidade com os estudos de Goodfellow, Bengio e Courville (2016, p. 168), uma camada densa é comumente usada em redes neurais, em que cada neurônio na camada está totalmente conectado a todos os neurônios na camada anterior. Os cálculos realizados em uma camada densa envolvem multiplicação de matriz entre a entrada dos dados e os pesos (parâmetros) da camada, seguida por uma função de ativação. Aqui estão os cálculos para uma camada densa: Seja \mathbf{x}_t a matriz de entrada de dimensão $(\mathbb{R}^{N \times M})$, em que N é o número de elementos e M é o número de variáveis observadas. Seja $\mathbf{W}_{\mathbf{iz}}$ a matriz de pesos da camada densa de dimensão $(\mathbb{R}^{M \times P})$, com P sendo o número de neurônios na camada densa. Além disso, seja $\mathbf{b}_{\mathbf{iz}}$ a matriz de viés da camada densa com dimensão $\mathbb{R}^{M \times P}$. A saída da camada densa \mathbf{Z} é calculada da seguinte forma:

$$\mathbf{Z} = \mathbf{W}_{\mathbf{iz}}\mathbf{x}_t + \mathbf{b}_{\mathbf{iz}},$$

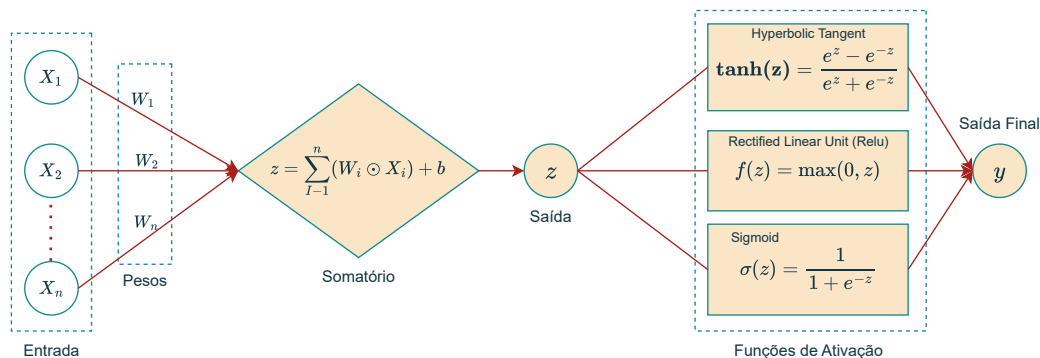
aqui, $\mathbf{W}_{\mathbf{iz}}\mathbf{x}_t$ representa a multiplicação de matriz entre os pesos ($\mathbf{W}_{\mathbf{iz}}$) e a entrada (\mathbf{x}_t) da camada densa, e $\mathbf{b}_{\mathbf{iz}}$ é o viés adicionado para produzir a saída final. É importante notar que após essa operação, geralmente é aplicada uma função de ativação aos elementos de \mathbf{Z} para introduzir não linearidade na camada densa conforme (Goodfellow; Bengio; Courville, 2016, p. 169).

3.5 Funções de Ativação

Funções de ativação, conforme Goodfellow, Bengio e Courville (2016, p. 170), são componentes fundamentais em redes neurais, responsáveis por introduzir não-linearidades nas saídas das camadas, o que permite às redes neurais aprender e modelar relações complexas nos dados. Essas funções não possuem parâmetros ajustáveis e são fixas, usadas especificamente para introduzir não-linearidade nas redes neurais conforme (Goodfellow;

Bengio; Courville, 2016). A Figura 4 ilustra a transformação linear e a ativação linear em uma camada densa final de uma rede neural.

Figura 4 – Ilustração da transformação linear e da ativação linear em uma camada densa da rede neural.



Fonte: Elaborada pelo autor, 2024.

Como está descrito na documentação do Keras (2024), entre as funções de ativação mais comuns estão mencionadas com todos os detalhes na mesma seção, a função *sigmoid*, a tangente hiperbólica *tanh* e a função *Rectified Linear Unit* (ReLU). A função *sigmoid*, também conhecida como função logística, é definida pela Equação 3.2:

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \quad (3.2)$$

em que z é a entrada da função. Essa função transforma números reais em um intervalo de valores entre 0 e 1, sendo utilizada principalmente em problemas de classificação binária.

A função tangente hiperbólica (\tanh), descrita pela Equação 3.3, também comprime valores reais, mas para um intervalo entre -1 e 1, o que pode auxiliar na convergência de alguns modelos:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \quad (3.3)$$

A função *Rectified Linear Unit* (ReLU) é uma função de ativação não linear que retorna zero para valores negativos e mantém valores positivos lineares, como mostrado na Equação 3.4:

$$f(z) = \max(0, z). \quad (3.4)$$

Conhecida por sua simplicidade e eficácia, a ReLU tem se mostrado especialmente útil para o treinamento de redes neurais profundas.

3.6 Algoritmo de Otimização *Adam*: Uma Visão Geral

O algoritmo *Adam*, proposto por Kingma e Ba (2017), utiliza médias móveis exponenciais dos gradientes para atualizar os parâmetros, acelerando a convergência e evitando que o modelo fique preso em mínimos locais. O *Adam* incorpora estimativas de primeira e segunda ordens com correções de viés para melhorar a eficácia da otimização. As configurações padrão para os problemas de aprendizado de máquina testados são $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ e $\epsilon = 10^{-8}$. Todas as operações em vetores são realizadas elemento a elemento (matricialmente). Com β_t^1 e β_t^2 denotados como β_1 e β_2 elevados à potência t . α : Tamanho do Passo (Padrão sugerido: 0.001); $\beta_1, \beta_2 \in [0, 1)$: Taxas de Decaimento Exponencial para as Estimativas de Momento; $f(\theta)$:

Função Objetiva Estocástica com Parâmetros θ ; g_t : É o vetor dos gradientes da função objetiva estocástica $f_t(\theta)$ em relação aos parâmetros θ , calculado no passo t ; θ_0 : Vetor de Parâmetros Inicial. Inicialização: $m_0 \rightarrow 0$ (Inicializar 1º Momento); $v_0 \rightarrow 0$ (Inicializar 2º Momento); $t \rightarrow 0$ (Inicializar Passo). De acordo com Kingma e Ba (2017, p.2), o algoritmo de *Adam* segue os seguintes passos:

Algoritmo 1: Pseudocódigo do Algoritmo de Otimização *Adam*.

```

1 while  $\theta_t$  não convergiu do
2    $t \leftarrow t + 1$ ;  $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$  (Obter gradientes em relação à função objetiva estocástica no passo  $t$ );
3    $m_t \leftarrow \beta_1 \times m_{t-1} + (1 - \beta_1) \times g_t$  (Atualizar estimativa viesada do 1º Momento);
4    $v_t \leftarrow \beta_2 \times v_{t-1} + (1 - \beta_2) \times g_t^2$  (Atualizar estimativa viesada do 2º Momento);
5    $\hat{m}_t \leftarrow \frac{m_t}{1 - \beta_1^t}$  (Calcular estimativa corrigida de viés do 1º Momento);
6    $\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_2^t}$  (Calcular estimativa corrigida de viés do 2º Momento);
7    $\theta_t \leftarrow \theta_{t-1} - \alpha \times \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$  (Atualizar parâmetros);
8   end while
9 return  $\theta_t$  (Parâmetros Resultantes);

```

3.6.1 Descrição dos parâmetros

De acordo com Kingma e Ba (2017), o algoritmo *Adam* é considerado uma técnica avançada de otimização que calcula taxas de aprendizado adaptativas para cada parâmetro. Ele combina características dos métodos Adagrad e RMSprop, mantendo médias móveis exponenciais dos gradientes e dos gradientes ao quadrado para ajustar as taxas de aprendizado.

m_t e v_t são estimativas do primeiro momento (a média) e do segundo momento (a variância não centralizada) dos gradientes, respectivamente, daí o nome do método. Como m_t e v_t são inicializados como vetores de zeros, os autores do *Adam* observam que eles são tendenciosos a valores próximos de zero, especialmente durante os passos iniciais, e particularmente quando as taxas de decaimento são pequenas [...] (Ruder, 2017, p.7).

1

Eles contrabalançam os vieses calculando estimativas corrigidas de viés para o primeiro e segundo momentos:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t,$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2,$$

em que, m_t representa a estimativa da média dos gradientes e v_t a estimativa da variância não centralizada. Para corrigir o viés de inicialização dessas estimativas, Kingma e Ba (2017) recomenda calcular as correções de viés da seguinte forma:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t},$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t},$$

com essas estimativas corrigidas, a atualização dos parâmetros é apresentada na Equação 3.5:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t. \quad (3.5)$$

¹ Texto original: m_t and v_t are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients respectively, hence the name of the method. As m_t and v_t are initialized as vectors of 0's, the authors of Adam observe that they are biased towards zero, especially during the initial time steps, and especially when the decay rates are small [...] (Ruder, 2017, p.7).

De acordo com Kingma e Ba (2017), *Adam* adapta-se a problemas de aprendizado de máquina em larga escala e de alta dimensão.

3.7 Regularização de Redes Neurais Recorrentes com *Dropout*

Conforme a documentação do Keras (2024), a implementação do *Dropout* desativa aleatoriamente uma fração dos neurônios durante o treinamento, a taxa de dropout, ajuda a prevenir o sobreajuste. As entradas ativas são escaladas para manter a soma inalterada. O *Dropout* é aplicado apenas em modo de treinamento, permitindo que o modelo aprenda representações mais robustas e generalizáveis, essenciais para evitar o sobreajuste em modelos complexos.

O *Dropout* é uma técnica popular para prevenir o sobreajuste durante o treinamento de redes neurais. [...] usando o *Dropout*, a ativação y^l na camada l é dada pela Equação 3.6, em que y^{l-1} é a entrada na camada l , W_l é o peso para a camada l , b é o bias, f é a função de ativação não linear (ou seja, ReLU) e r é uma máscara binária, onde cada entrada é retirada de uma distribuição Bernoulli(p) com probabilidade p de ser 1. [...] o fator $\frac{1}{1-p}$ usado durante o treinamento garante que, no momento do teste, quando nenhuma unidade é desativada, a entrada total correta alcançará cada camada, podemos observar essas descrições na Equação 3.6

$$y^l = f\left(\frac{1}{1-p} \times W_l \times (r^{l-1} \times y^{l-1}) + b_l\right) \quad (3.6)$$

(Sainath *et al.*, 2013, p.2).

2

3.8 Análise de Sazonalidade dos Focos de Calor no Pantanal

A decomposição de séries temporais é uma técnica essencial para a análise de dados temporais, pois permite separar a série em componentes que fornecem uma visão mais clara sobre o comportamento dos dados. No contexto da análise de séries temporais, os componentes principais são a tendência, a sazonalidade e o resíduo. Esta seção detalha esses componentes e discute o método *Seasonal-Trend decomposition using LOESS (STL)* (Cleveland *et al.*, 1990).

3.8.1 Componentes de uma Série Temporal

Para uma série temporal aditiva (Equação 3.7), a decomposição pode ser representada da seguinte forma:

$$Y_t = T_t + S_t + R_t, \quad (3.7)$$

em que Y_t é o valor observado da série no tempo t , T_t é o componente de tendência, S_t é o componente sazonal, e R_t é o resíduo, que representa as flutuações não explicadas pela tendência ou sazonalidade (Cleveland *et al.*, 1990; Statsmodels, 2024).

² Tradução de: Dropout is a popular technique to prevent over-fitting during neural network training. [...] using dropout the activation y^l at layer l is given by Equation 9, where y^{l-1} is the input into layer l , W_l is the weight for layer l , b is the bias, f is the non-linear activation function (i.e., ReLU) and r is a binary mask, where each entry is drawn from a Bernoulli(p) distribution with probability p of being 1. Since dropout is not used during decoding, the factor $\frac{1}{1-p}$ used during training ensures that at test time, when no units are dropped out, the correct total input will reach each layer

$$y^l = f\left(\frac{1}{1-p} \times W_l \times (r^{l-1} \times y^{l-1}) + b_l\right)$$

(Sainath *et al.*, 2013, p.2).

3.8.2 Cálculo da Tendência, Sazonalidade e Resíduo

A tendência (Equação 3.8), representa a direção geral da série (crescente, decrescente ou estável) e pode ser estimada por meio de uma média móvel de janela m (por exemplo, $m = 12$ para dados mensais de um ano). A média móvel suaviza as flutuações de curto prazo e destaca a tendência de longo prazo (Statsmodels, 2024):

$$T_t = \frac{1}{m} \sum_{i=t-(m/2)}^{t+(m/2)} Y_i. \quad (3.8)$$

A sazonalidade (Equação 3.9), por sua vez, captura padrões cíclicos que se repetem em intervalos regulares. Para estimar a componente sazonal, subtrai-se a tendência da série original, e a média dos resíduos em cada ponto da janela de sazonalidade que é calculada (Cleveland *et al.*, 1990):

$$S_t = \frac{1}{n/m} \sum_{k=1}^{n/m} (Y_{t+km} - T_{t+km}), \quad (3.9)$$

em que n é o número total de observações, m é o período da sazonalidade, k é o índice de iteração e $t + km$ indica as posições na série que estão separadas por um intervalo sazonal completo. O resíduo é obtido subtraindo-se a soma dos componentes de tendência e sazonalidade do valor observado em cada ponto:

$$R_t = Y_t - (T_t + S_t). \quad (3.10)$$

O resíduo (Equação 3.10), representa as flutuações que não podem ser explicadas pelos componentes de tendência e sazonalidade (Cleveland *et al.*, 1990; Statsmodels, 2024).

Figura 5 – Análise de sazonalidade, tendência e resíduos dos focos de calor no Pantanal (1998 - 2024).



Fonte: Elaborada pelo autor, 2024.

A Figura 5 foi criada com o pacote `statsmodels`, Statsmodels (2024), na linguagem de programação Python (2024), e foi configurado o modelo aditivo com uma periodicidade de 12 *lags*, que buscará padrões que se repetem a cada 12 meses, assim foi identificado detalhes na decomposição sazonal na série temporal dos focos de calor na região do Pantanal que apresentou um padrão anual.

4 MATERIAL E MÉTODOS

Nesta seção, descreve-se o procedimento para modelar e prever séries temporais utilizando redes neurais recorrentes. Foram utilizados dados de contagem dos focos de calor detectados pelo satélite *AQUA_M-T* no Bioma do Pantanal, abrangendo uma série histórica registrada de junho de 1998 a setembro de 2024. Esses dados estão disponíveis no site do INPE (2024).

O processo metodológico para modelagem e previsão da série temporal segue práticas estabelecidas na literatura de séries temporais e aprendizado de máquina, conforme descrito por Goodfellow, Bengio e Courville (2016) e nas documentações de Keras (2024), Scikit-learn (2024) e TensorFlow (2024), que detalham os procedimentos descritos a seguir. Primeiramente, de acordo com a documentação de TensorFlow (2024), os dados foram divididos em conjuntos de treino, validação e teste para avaliar a performance do modelo. Foram aplicadas técnicas como validação cruzada, que, no contexto de séries temporais, ocorreu em três conjuntos, com inicializações diferentes fixadas por sementes, garantindo a robustez da validação e a reprodutibilidade dos resultados. Cada semente foi previamente definida com os valores 2030, 2040 e 2050. Após garantir que o modelo apresentava uma boa capacidade de generalização, de acordo com Goodfellow, Bengio e Courville (2016, p. 12), optou-se por treinar o modelo final utilizando todos os dados disponíveis. Tal escolha foi realizada após determinar o modelo que apresentou o menor erro absoluto médio no conjunto de teste, previamente separado para comparação.

A abordagem visa maximizar a precisão das previsões, especialmente em cenários de passos à frente da última observação treinada. De acordo com Géron (2017), no contexto de *deep learning*, em que ajustes finos (*fine-tuning*) são comuns, o uso do conjunto completo de dados após a validação é uma prática justificada para aprimorar o desempenho, conforme discutido também por Goodfellow, Bengio e Courville (2016, p. 12). Nesse sentido, após escolher o melhor modelo na etapa de treino, validação e teste, utilizou-se o modelo que melhor generalizou para o conjunto de teste e, após treinado com 100% dos dados, realizaram-se previsões de 12 passos à frente da série temporal.

4.1 Preparação dos Dados

Na preparação dos dados, adotou-se uma abordagem de treino, validação e teste, adaptada para séries temporais contínuas, conforme descrito na documentação de TensorFlow (2024), que especifica claramente como implementar de forma correta, utilizando a linguagem de programação (Python, 2024), modelos de aprendizado de máquina e aprendizado profundo em problemas de séries temporais. Para garantir a eficácia da avaliação do modelo, o processo de divisão dos dados foi realizado em blocos. Primeiramente, removeram-se os últimos 12 *lags* (meses) da série para o conjunto de teste, considerando a série completa menos esses 12 *lags*. Em seguida, removeram-se 24 *lags* adicionais para o conjunto de validação, resultando na série completa menos 36 *lags* para o treinamento.

Embora tenha sido seguida a abordagem de divisão de dados, utilizou-se validação cruzada com três sementes (2030, 2040 e 2050) para avaliar a performance do modelo. Conforme descrito por Géron (2017), a validação cruzada é essencial para garantir que o modelo generalize bem para novos dados. Utilizaram-se três sementes distintas para criar três conjuntos de inicializações diferentes, o que permitiu avaliar a robustez e a estabilidade do modelo com diferentes conjuntos de pesos e, conseqüentemente, diferentes resultados nas previsões dos conjuntos de treino, validação e teste.

A divisão final dos dados foi a seguinte: o conjunto de treino foi de junho de 1998 a setembro de 2021, o conjunto de validação foi de outubro de 2021 a setembro de 2023 e o conjunto de teste foi de outubro de 2023 a setembro de 2024. Essa abordagem, didaticamente explicada no tutorial da documentação Scikit-learn (2024), combinada com o uso de sementes fixas, garante a replicabilidade dos resultados e confirma a consistência das generalizações para os conjuntos de treino, validação e teste.

4.2 Configuração dos Modelos

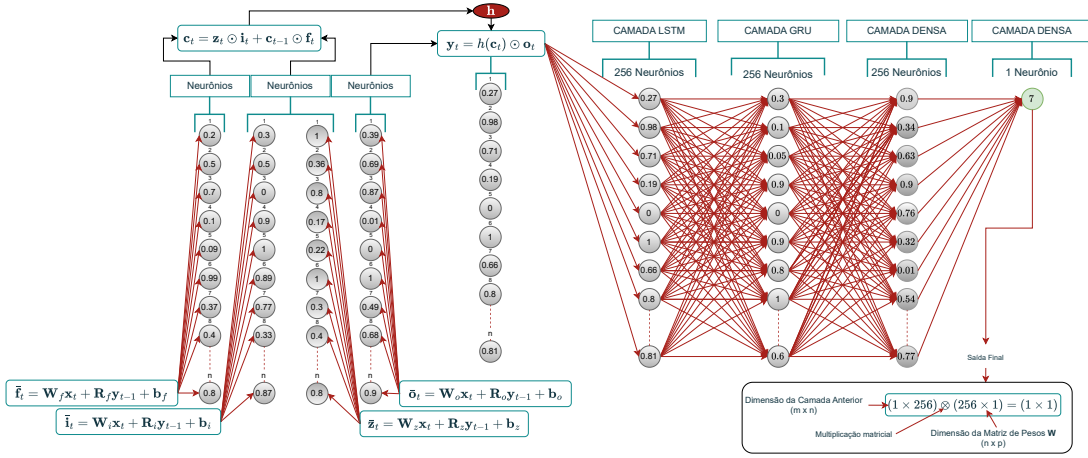
Foi utilizada a combinação dos modelos de redes neurais recorrentes LSTM e GRU. Diversas pesquisas demonstraram a eficácia dessa abordagem em problemas de séries temporais, especialmente em contextos relacionados a dados ambientais. Estudos como os de Gao *et al.* (2020), Lindemann *et al.* (2021), Wang *et al.* (2023), Mousa *et al.* (2024), Ahmad, Azhar e Satter (2024) e Tavares e Olinda (2024), corroboram que a integração dessas arquiteturas é benéfica para a modelagem e previsão de séries temporais, mostrando resultados que convergem na captura de padrões complexos em dados ambientais.

- **Camada de Entrada:** Recebe os dados da série temporal em janelas previamente fixadas. Escolhemos o tamanho da janela como 12, para que, a partir de 12 meses, possamos prever o valor do 13º ponto. Essa janela é "móvel", criando diversos blocos de tamanho 12 até os 12 últimos valores da série histórica, garantindo a estruturação correta para a entrada de informações na rede neural. Cada bloco, ao passar pela arquitetura, gera uma previsão para um ponto à frente do bloco, seguindo passo-a-passo até o fim da série.
- **Camada Recorrente:** O modelo utiliza uma combinação de LSTM e GRU, configuradas em sequência. Primeiro, uma camada LSTM com 256 neurônios é utilizada, permitindo que a saída de cada passo temporal (com 256 neurônios) seja passada diretamente para a próxima camada. Foi aplicada uma taxa de *dropout* de 20% nesta camada para ajudar a prevenir o *overfitting*.
- **Camada GRU:** Recebe a saída da camada LSTM e processa-a com seus 256 neurônios. A GRU refina as informações temporais capturadas pela LSTM, proporcionando uma representação mais rica e reduzindo potenciais redundâncias. Assim como na camada anterior, uma taxa de *dropout* de 20% foi aplicada aqui.
- **Camada Densa com Ativação ReLU:** Após a camada GRU, uma camada totalmente conectada (densa) com 256 neurônios e função de ativação ReLU é utilizada, proporcionando uma transformação não-linear que ajuda a modelar relações complexas nos dados. Uma taxa de *dropout* de 20% também foi aplicada nesta camada.
- **Camada de Saída:** Finalmente, a saída da camada densa passa para uma camada linear com 1 neurônio e uma restrição *kernel_constraint = NonNeg()*. Essa restrição assegura que todos os pesos desta camada sejam positivos, atribuindo valor zero para quaisquer pesos negativos, de forma a fornecer a previsão final para cada janela de entrada, garantindo um resultado positivo.

A Figura 6 ilustra a arquitetura da rede neural, composta pelas seguintes camadas: uma camada LSTM com 256 neurônios, uma camada GRU com 256 neurônios, uma camada densa com 256 neurônios e uma camada densa de saída com 1 neurônio.

A arquitetura da Figura 6 é explicada da seguinte forma: o vetor X_t no tempo t é processado pela camada LSTM com 256 neurônios, resultando em uma saída de dimensão (256×1) , pois nosso trabalho envolve uma

Figura 6 – Esquema detalhado da Rede Neural Recorrente híbrida (LSTM+GRU) com camadas densas.



Fonte: Elaborada pelo autor, 2024.

única variável ao longo do tempo (focos de calor). Essa saída serve como entrada para a camada GRU, que também possui 256 neurônios. A saída da GRU é então processada por uma camada densa com 256 neurônios, mantendo a dimensão (256 × 1). Finalmente, essa saída é alimentada na camada final com ativação linear de 1 neurônio, gerando uma previsão única para o próximo ponto do vetor de entrada. Para ilustrar, considere um vetor de dados com 12 valores, em que X_t corresponde aos valores de 1 a 12, prevendo o 13º valor. Esse processo é repetido ao mover o vetor de entrada: o segundo vetor será de 2 a 13, prevendo o 14º ponto, e assim por diante até o final da série. A Figura 6 apresenta a previsão de apenas um vetor. No funcionamento completo, o modelo computa todos os vetores de dimensões pré-definidas, resultando em previsões para cada vetor X_{t_i} de acordo com o tamanho da série. Assim, para valores de 1 a n , os dados são organizados em *arrays* de tamanho 12, com X_{t1} contendo do 1º ao 12º elemento, X_{t2} do 2º ao 13º, e assim sucessivamente.

4.3 Treinamento e Avaliação dos Modelos

Os modelos foram treinados utilizando a linguagem de programação Python (2024), com as bibliotecas Keras (2024) e TensorFlow (2024). O treinamento foi realizado por 500 *epochs*, utilizando a métrica de erro absoluto médio (MAE) para avaliar o desempenho. A métrica MAE é calculada pela Equação 4.1:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (4.1)$$

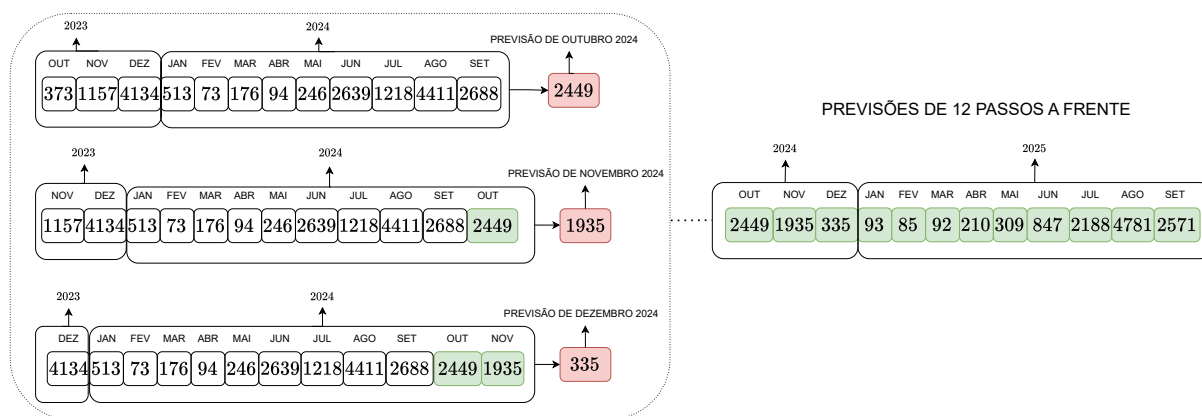
em que n é o número total de blocos, y_i representa os valores reais da série temporal e \hat{y}_i representa as previsões geradas pelo modelo. A MAE mede a média dos erros absolutos entre as previsões e os valores reais, fornecendo uma indicação clara do desempenho do modelo.

Avaliação das Previsões: As previsões geradas para cada época foram comparadas com os dados reais da série temporal. A precisão das previsões foi avaliada através da métrica MAE, que calcula a média dos erros absolutos das previsões. Os parâmetros iniciais dos modelos foram definidos utilizando uma distribuição de probabilidade específica, a distribuição normal de He, que é definida com média zero e desvio padrão $\sqrt{2/n}$, com n sendo o número de blocos na camada de entrada. Essa escolha visa garantir uma inicialização adequada dos pesos, facilitando o treinamento eficaz de redes neurais profundas, conforme descrito na documentação do *Keras*.

4.4 Como as Previsões são Calculadas?

As previsões foram realizadas após o treinamento completo dos dados de focos de calor registrados na região do Pantanal, Brasil, disponíveis no site do INPE (2024), abrangendo o período de junho de 1998 até setembro de 2024. Após o treinamento, utilizamos a função `predict` do pacote Keras (2024), para gerar as previsões. O modelo treinado, que já contém todos os parâmetros otimizados e ajustados, foi armazenado e salvo como o mais acurado durante o processo de treinamento. Este modelo foi configurado para trabalhar com uma variável e estruturado em blocos de tamanho 12. A função `predict` segue a sequência dos dados, incorporando as previsões de cada bloco para gerar novos resultados, adicionando esses resultados à série temporal e prevendo o próximo ponto. A ilustração detalhada desse processo está apresentada na Figura 7.

Figura 7 – Previsão de focos de calor: funcionamento do `predict` em redes neurais recorrentes.



Fonte: Elaborada pelo autor, 2024.

O processo é detalhado a seguir: a função `predict` utiliza o último bloco de 12 observações da série temporal, que vai de outubro de 2023 a setembro de 2024, para prever o 13º ponto correspondente a outubro de 2024. Utilizamos a abordagem de "janela deslizante" que está apresentada na documentação do TensorFlow (2024), isso nos permite interpretar que após a primeira previsão para outubro de 2024, o modelo integre essa previsão e gere uma nova previsão para o próximo mês. No segundo passo, por exemplo, ele utiliza as observações de novembro de 2023 a outubro de 2024, agora incluindo a previsão anterior (de outubro), para prever novembro de 2024. No terceiro passo, o modelo considera os dados de dezembro de 2023 a novembro de 2024, integrando as previsões obtidas de outubro e novembro à série temporal, e assim por diante. Esse processo se repete até que todas as previsões dos 12 meses sejam realizadas. Essa abordagem assegura que cada previsão mensal se baseie nos dados históricos mais recentes, juntamente com as previsões feitas nos passos anteriores, resultando em uma modelagem robusta para séries temporais de dados de contagem.

5 RESULTADOS E DISCUSSÕES

Nesta seção, por meio da base de dados, apresenta-se as principais estatísticas descritivas da série temporal de focos de calor no Pantanal, destacando estatísticas como média, mediana e valores máximos e mínimos ao longo do tempo. Em seguida, ajustam-se três modelos de redes neurais recorrentes avaliados: *Long Short-Term Memory* (LSTM), *Gated Recurrent Unit* (GRU) e uma abordagem híbrida que combina ambos (LSTM+GRU). Para avaliar o desempenho de cada modelo, utilizou-se a métrica do erro absoluto médio (MAE), aplicada aos conjuntos de treino, validação e teste. Os resultados estão organizados em tabelas e gráficos que detalham o desempenho dos modelos em termos de erro para cada conjunto de dados.

5.1 Análise Descritiva dos focos de calor no Pantanal

A Tabela 1 apresenta a frequência com que os valores mínimos e máximos de focos de calor foram registrados na região do Pantanal ao longo de 27 anos, abrangendo o período de junho de 1998 até setembro de 2024.

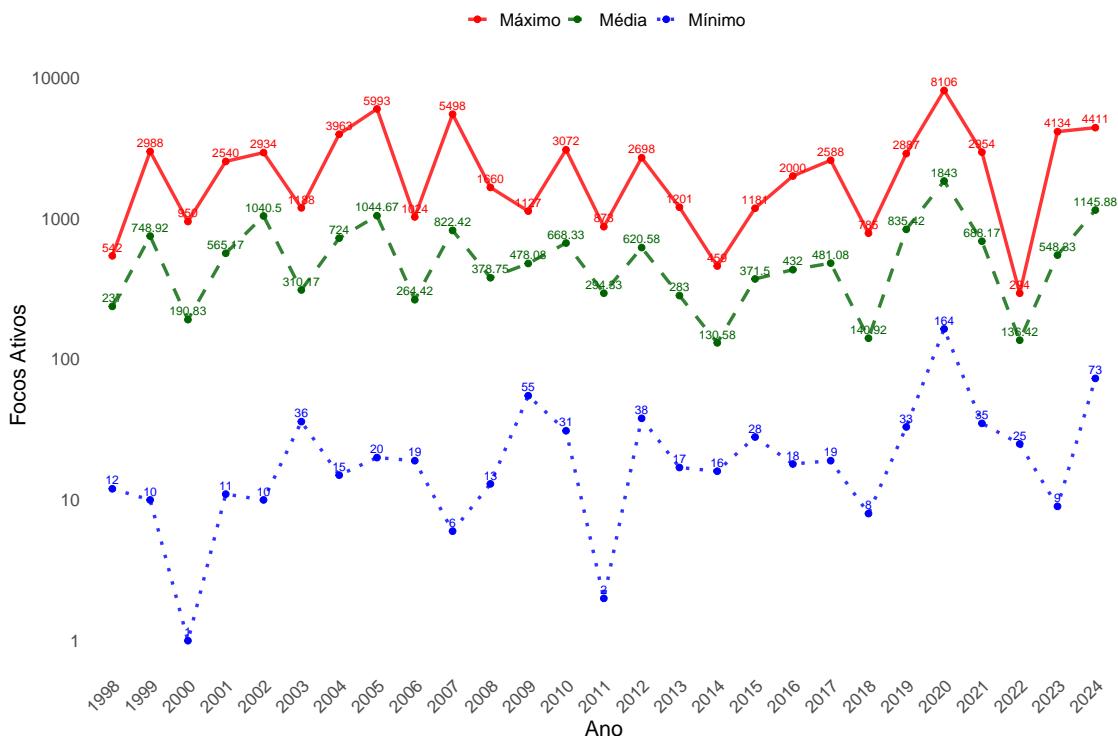
Tabela 1 – Estatísticas descritivas da série temporal anual dos focos de calor no Pantanal.

Ano	Valor Mínimo	Valor Médio	Valor Máximo
1998	12	237,00	542
1999	10	748,92	2988
2000	1	190,83	950
2001	11	565,17	2540
2002	10	1040,50	2934
2003	36	310,17	1188
2004	15	724,00	3963
2005	20	1044,67	5993
2006	19	264,42	1024
2007	6	822,42	5498
2008	13	378,75	1660
2009	55	478,08	1127
2010	31	668,33	3072
2011	2	294,33	873
2012	38	620,58	2698
2013	17	283,00	1201
2014	16	130,58	459
2015	28	371,50	1181
2016	18	432,00	2000
2017	19	481,08	2588
2018	8	140,92	785
2019	33	835,42	2887
2020	164	1843,00	8106
2021	35	688,17	2954
2022	25	136,42	294
2023	9	548,33	4134
2024	73	1145,88	4411

Fonte: Elaborada pelo autor, 2024, com base nos dados do *Instituto Nacional de Pesquisas Espaciais (INPE)*.

A análise dos dados de focos de calor no Pantanal revela que os valores mínimos registrados tendem a ocorrer sempre no primeiro semestre do ano, exclusivamente entre dezembro e junho, de acordo com o histórico de 27 anos também observados na Figura 8.

Figura 8 – Distribuição anual: valores mínimos, máximos e média dos focos de calor no Pantanal (1998 - 2024).



Fonte: Elaborada pelo autor, 2024, com base nos dados do *Instituto Nacional de Pesquisas Espaciais (INPE)*.

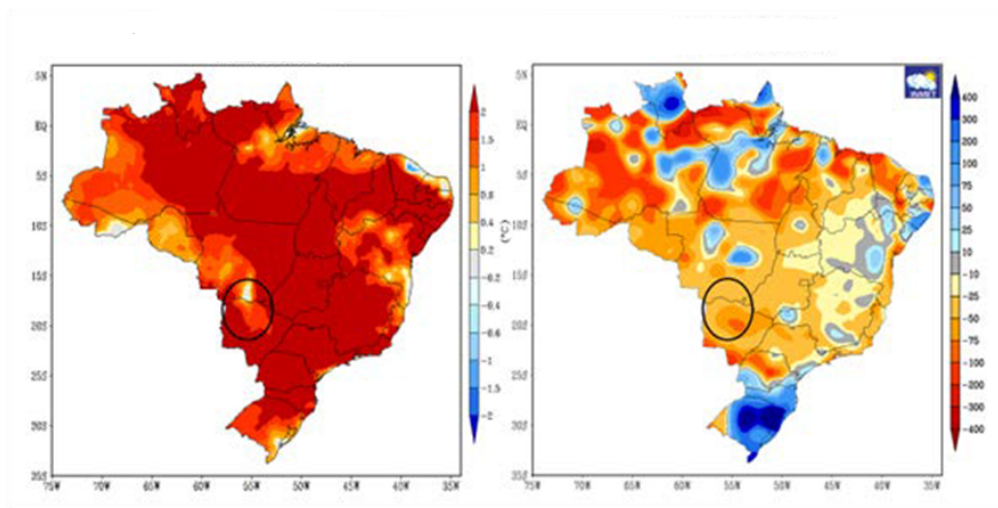
Por outro lado os valores máximos concentraram-se de julho a novembro, com alerta de altos picos principalmente nos meses de agosto e setembro que se destacam como os períodos de maior incidência, observa-se o mês de setembro repetindo-se 16 vezes como o de maior pico anual. Esse padrão sazonal evidencia que o segundo semestre é o período mais crítico, com maior intensidade de focos de calor, especialmente nos meses de agosto e setembro, alertando para os riscos nessa época do ano.

5.1.1 Análise dos Valores Extremo dos focos de calor no Pantanal

Conforme o *Plano de Ação para Prevenção e Controle do Desmatamento e das Queimadas no Pantanal*, do Ministério do Meio Ambiente e Mudança do Clima (2024), a incidência de incêndios no Pantanal tem sido agravada por condições climáticas extremas, incluindo secas prolongadas e ondas de calor. Em 2024, entre janeiro e setembro, foram registrados quase 12 mil focos de calor, sendo o maior índice já registrado nesse período, ainda que, em julho e agosto, o total de focos tenha sido menor que os observados em 2020 — ano de alta incidência de queimadas. Essas condições climáticas desfavoráveis, somadas à baixa precipitação e aos baixos níveis das águas do Rio Paraguai, demonstram que o ciclo de incêndios tende a se repetir, especialmente no segundo semestre, o que é consistente com os dados históricos apresentados. A Figura 9 apresenta uma

maior propensão a focos de calor durante a estação seca, que ocorre predominantemente de julho a novembro, indicando a importância de ações preventivas concentradas nesse período.

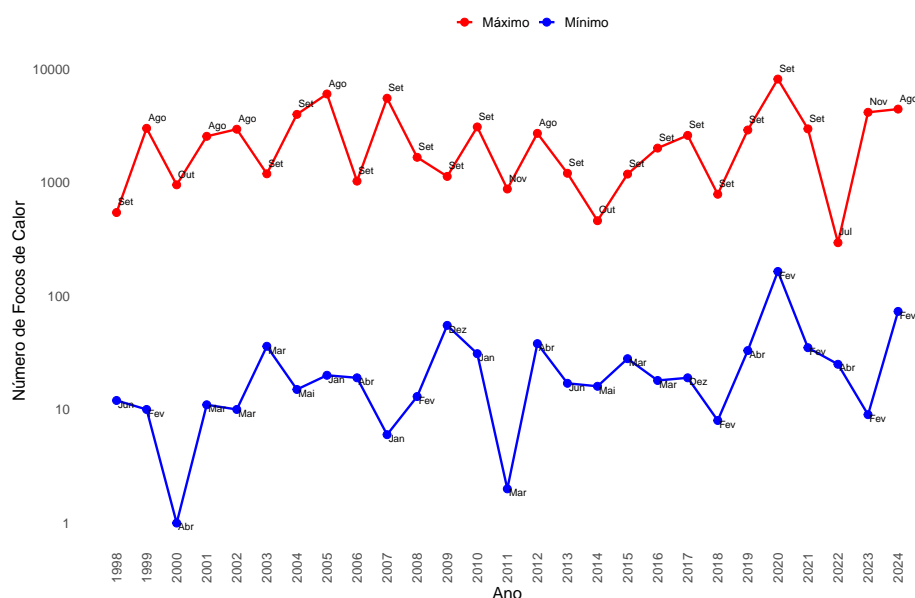
Figura 9 – Anomalias de temperatura e precipitação observadas no Brasil, com destaque para a região do Pantanal.



Fonte: Instituto Nacional de Meteorologia (Inmet), 2024, apud (Ministério do Meio Ambiente e Mudança do Clima, 2024).

A partir dessas análises podemos concluir que, o segundo semestre concentra a maior parte dos focos de calor máximos, especialmente nos meses de agosto e setembro, que registraram 6 e 16 ocorrências, respectivamente ao longo de 27 anos analisados. A Figura 10 a seguir ilustra os valores mínimos e máximos dos focos de calor por ano, junto com os meses correspondentes.

Figura 10 – Valores Mínimos e Máximos de focos de calor no Pantanal (1998 - 2024)



Fonte: Elaborada pelo autor, 2024, com base nos dados do Instituto Nacional de Pesquisas Espaciais (INPE).

Essa visualização facilita a identificação de padrões sazonais e auxilia na compreensão da dinâmica dos focos de calor no Pantanal ao longo dos anos.

5.1.2 *Preservação da Estrutura Sazonal em Dados de Contagem na Modelagem dos focos de calor no Pantanal*

A modelagem foi realizada utilizando dados originais, sem transformação para uma escala padronizada, pois essa abordagem se mostrou mais eficaz. Segundo Benidis *et al.* (2022), na seção 3.9.3 *Problem characteristics*, a natureza da previsão (probabilística versus pontual) influencia a escolha do modelo. Para previsões em que a distribuição completa é crucial, é essencial que os dados reflitam a realidade original, evitando normalizações que possam obscurecer a variabilidade. A normalização pode diluir a precisão dos dados de contagem, que são inteiros, e manter os dados na forma original permite que o modelo capture melhor as variações. Além disso, conforme mencionado por Benidis *et al.* (2022), a importância dos valores históricos para previsões futuras sugere que a manutenção dos dados originais é vital para modelar corretamente as relações de dependência temporal. Transformações que alterem esses dados podem desestabilizar essas relações e prejudicar a previsão concreta. No contexto da modelagem dos focos de calor, os dados são de contagem e não devem ser tratados como valores probabilísticos. Descartamos, portanto, a complexidade de transformar valores inteiros em conjuntos probabilísticos decimais, o que seria excessivamente complexo.

Um aspecto importante de um modelo é a sua natureza de previsão, ou seja, se ele produz previsões probabilísticas ou pontuais. A escolha disso depende muito da aplicação subjacente. [...] Nesse caso, um modelo que se concentra em um quantil específico de importância é mais interessante do que um modelo que prevê toda a distribuição com uma precisão possivelmente pior no quantil selecionado. [...] Os modelos de espaço de estados mostraram-se robustos quando há observações faltantes e/ou ruidosas (Benidis *et al.*, 2022, p. 22).

1

Neste ponto citado, podemos interpretar que o autor enfatiza que os modelos mais robustos para dados probabilísticos (transformados e/ou normalizados) são aqueles que predominantemente apresentam muito ruído, conseqüentemente uma natureza que não demonstra um padrão existente (sazonalidade). Assim, fica mais claro entender que conhecer os dados em questão traz mais confiabilidade na aplicação dos métodos.

5.2 **Análise das Métricas de Desempenho dos Modelos LSTM e GRU e Abordagem Híbrida que Combina Ambos (LSTM+GRU)**

A partir dos resultados de métricas apresentados, verificamos que o modelo híbrido (LSTM+GRU) apresentou menor erro absoluto médio ao comparado com os modelos LSTM e GRU isolados. Como podemos verificar, uma referência importante é que no estudo de Tavares e Olinda (2024), podemos verificar os resultados de modelos mistos apresentando menores erros de métrica também na perspectiva da utilização de dados de contagem sem transformações e/ou normalizações.

Este estudo avaliou o desempenho de redes neurais recorrentes, combinando as arquiteturas Long Short-Term Memory (LSTM) e Gated Recurrent Unit (GRU), na previsão de focos de calor detectados pelo satélite *AQUA_M-T* na Amazônia. A análise demonstrou que essa combinação é eficaz na captura de padrões temporais complexos, com a escolha final do modelo baseada no Erro Absoluto Médio (MAE). As previsões geradas por modelos combinados apresentaram um desempenho elevado, especialmente em séries temporais com forte sazonalidade, como é o caso da série histórica dos focos de calor na Amazônia (Tavares; Olinda, 2024, p. 15).

¹ Tradução de: One important aspect of a model is its forecast nature, i.e., if it produces probabilistic or point forecasts. The choice of this is highly dependent on the underlying application. [...] In this case, a model that focuses on a particular quantile of importance is of higher interest than a model that predicts the whole distribution with possibly worse accuracy on the selected quantile. [...] The choice of model should reflect the specific needs of the application (Benidis *et al.*, 2022, p. 22).

Os autores Aburass, Dorgham e Shaqsi (2024) demonstram que a modelagem com redes neurais profundas, incluindo a combinação de várias arquiteturas, é mais eficiente do que o uso de combinações menores ou isoladas. Entre essas arquiteturas compostas estão presentes as LSTM e GRU, que são relevantes para os dados que estamos modelando. Por essa razão, é essencial compreender como cada modelo se comporta em cenários nos quais a complexidade dos dados é significativa.

O modelo de ensemble LSTM + BiLSTM + CNN + GRU + GloVe apresenta o menor MSE. [...] tanto no treinamento quanto na validação, refletindo sua forte precisão preditiva e capacidades de generalização [...] (Aburass; Dorgham; Shaqsi, 2024, p. 6).

2

Portanto nos modelos apresentados o que demonstrou menores erros, considerando a métrica do erro absoluto médio (MAE) em todos os conjuntos analisados, tanto no conjunto de treinamento quanto na generalização do teste, foi o modelo híbrido (LSTM+GRU). Esses resultados indicam que o modelo híbrido é o mais eficiente para realizar previsões na modelagem de focos de calor no Bioma do Pantanal, os resultados são apresentados nas Tabelas 2 e 3.

Tabela 2 – Desempenho dos modelos em termos do erro absoluto médio (MAE) em dados de treinamento e teste para sementes de 2030, 2040 e 2050.

Modelo	Treino			Teste		
	Semente 2030	Semente 2040	Semente 2050	Semente 2030	Semente 2040	Semente 2050
LSTM	183,097	179,201	151,011	1150,667	1060,833	1002,0
GRU	198,056	192,825	191,041	1070,25	983,583	1082,833
(LSTM+GRU)	231,675	229,720	87,078	784,583	1064,417	1097,0

Fonte: Elaborada pelo autor, 2024.

Tabela 3 – Valores reais e preditos no conjunto de teste do modelo híbrido considerando o erro absoluto médio (MAE) correspondente à semente 2030.

Mês	Reais	Previstos	MAE	Ajuste
Outubro 2023	1157	1055	$ 1157 - 1055 = 102$	Subestimou
Novembro 2023	4134	2662	$ 4134 - 2662 = 1472$	Subestimou
Dezembro 2023	513	2137	$ 513 - 2137 = 1624$	Superestimou
Janeiro 2024	310	183	$ 310 - 183 = 127$	Subestimou
Fevereiro 2024	73	90	$ 73 - 90 = 17$	Superestimou
Março 2024	176	89	$ 176 - 89 = 87$	Subestimou
Abril 2024	94	87	$ 94 - 87 = 7$	Subestimou
Mai 2024	246	87	$ 246 - 87 = 159$	Subestimou
Junho 2024	2639	91	$ 2639 - 91 = 2548$	Subestimou
Julho 2024	1218	929	$ 1218 - 929 = 289$	Subestimou
Agosto 2024	4411	1720	$ 4411 - 1720 = 2691$	Subestimou
Setembro 2024	2688	2980	$ 2688 - 2980 = 292$	Superestimou
Métrica	$MAE = \frac{\sum Reais - Previsto }{12} = \frac{12.110}{12} = 784,58$			

Fonte: Elaborada pelo autor, 2024.

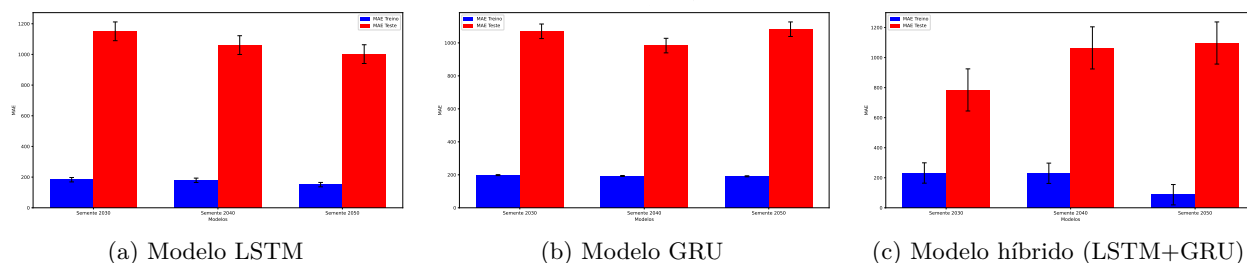
Na tabela 3 são apresentados os valores reais do conjunto de teste, os valores previstos pelo modelo híbrido (LSTM+GRU) e o cálculo do erro absoluto médio (MAE) para a semente 2030. Este conjunto de dados

² Tradução de: The ensemble model LSTM + BiLSTM + CNN + GRU + GloVe showcases the lowest MSE. [...] training and validation, reflecting its strong predictive accuracy and generalization capabilities [...] (Aburass; Dorgham; Shaqsi, 2024, p. 6).

reflete a generalização do modelo preditivo em 12 passos à frente, abrangendo o período de outubro de 2023 a setembro de 2024.

As Figuras 11a, 11b e 11c apresentam a comparação das métricas de desempenho para os modelos LSTM, GRU e híbrido (LSTM+GRU), considerando as sementes 2030, 2040 e 2050.

Figura 11 – Comparação das métricas de desempenho dos Modelos.



Fonte: Elaborada pelo autor, 2024.

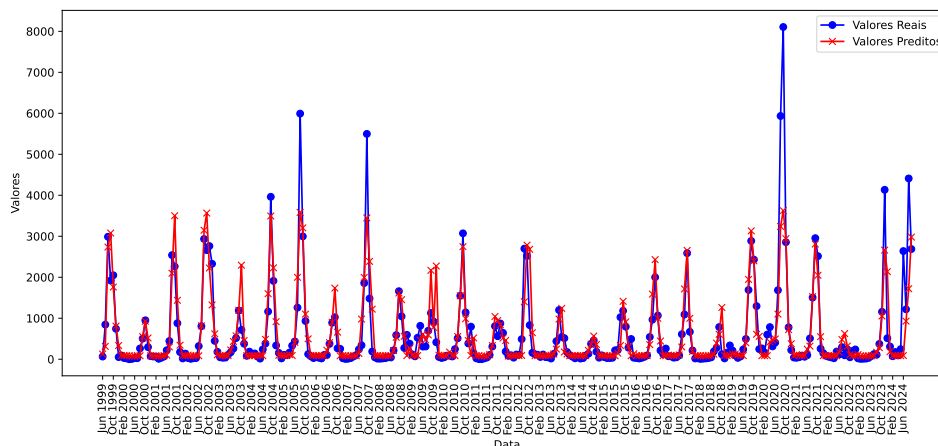
5.3 Análise de Desempenho dos Conjuntos de Treino, Validação e Teste

Nesta seção, realizaremos uma análise detalhada do desempenho do modelo preditivo híbrido LSTM+GRU utilizando a semente 2030, que apresentou o menor erro absoluto médio (MAE) no conjunto de teste, como apresentado e discutido na seção 5.2. Serão apresentados gráficos que ilustram os resultados obtidos nos diferentes conjuntos de dados: treino, validação e teste.

5.3.1 Conjunto de Treino

A Figura 12 representa o conjunto de treino utilizado no modelo híbrido, é neste conjunto que o modelo aprende a relacionar os dados de entrada com as saídas esperadas. O desempenho observado nesta fase é um indicativo inicial da capacidade preditiva do modelo em captar padrões.

Figura 12 – Conjunto de treino.

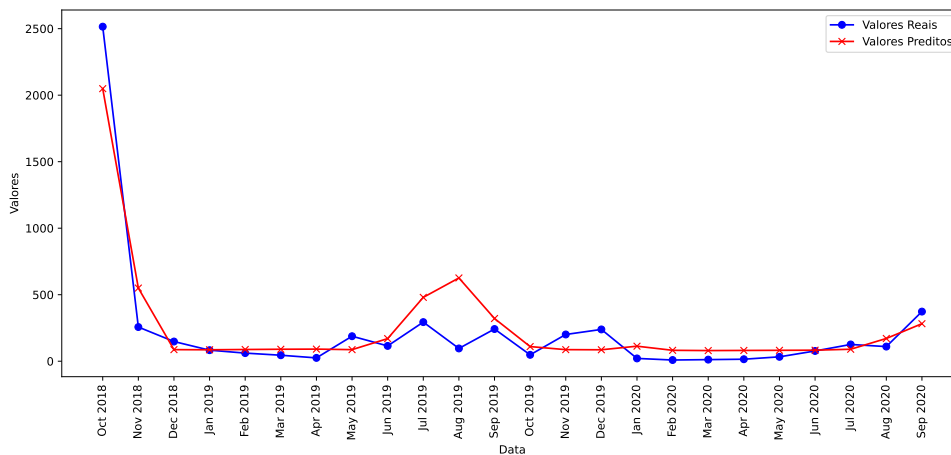


Fonte: Elaborada pelo autor, 2024.

5.3.2 Conjunto de Validação

A Figura 13 mostra o conjunto de validação. Este conjunto é utilizado para ajustar os hiperparâmetros do modelo e evitar o *overfitting* (ajuste excessivo) no conjunto de treinamento. A análise dos resultados nesta fase ajuda a garantir que o modelo não apenas memorize os dados de treino, mas generalize bem para novos dados.

Figura 13 – Conjunto de validação.

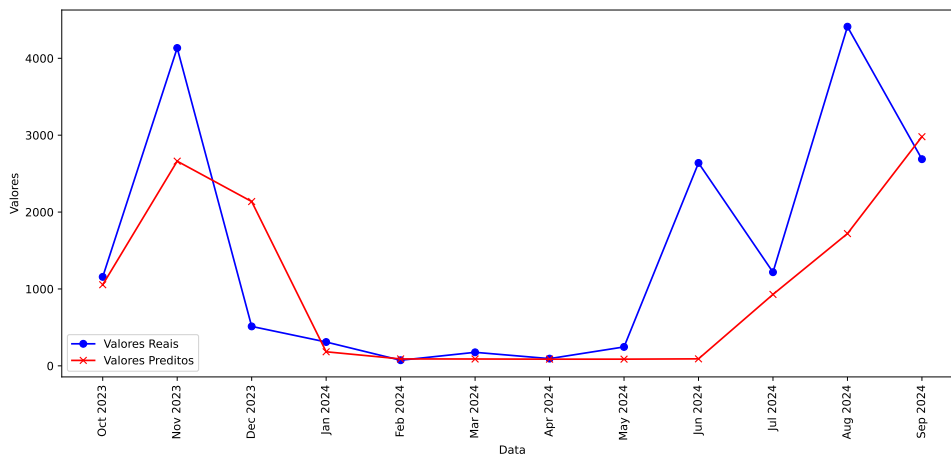


Fonte: Elaborada pelo autor, 2024.

5.3.3 Conjunto de Teste

A Figura 14 ilustra o conjunto de teste, que é usado para avaliar o desempenho final do modelo após o treinamento e validação. Os resultados obtidos nesta fase são fundamentais para medir a eficácia do modelo em previsões com dados que nunca foram vistos anteriormente.

Figura 14 – Conjunto de teste.

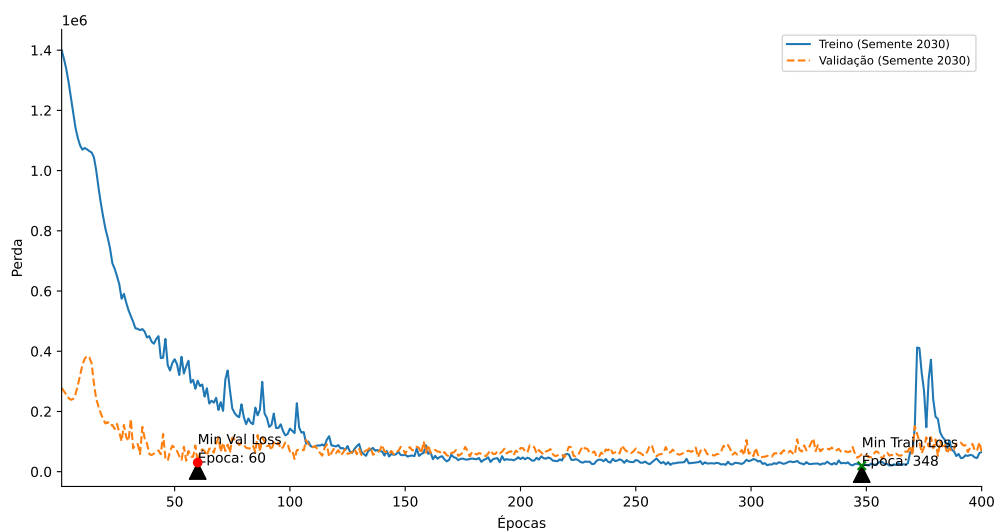


Fonte: Elaborada pelo autor, 2024.

5.3.4 Histórico de Perda

A Figura 15 apresenta o histórico de perda durante o treinamento e validação do modelo. O ponto de menor erro no conjunto de validação foi obtido na época 60, enquanto o conjunto de treino atingiu seu menor erro na época 348. Para a geração das previsões no conjunto de teste, são utilizados os parâmetros correspondentes ao menor erro encontrado no conjunto de validação.

Figura 15 – Histórico de perda (*Loss*) durante o treinamento e validação.



Fonte: Elaborada pelo autor, 2024.

5.4 Resultados das Previsões Considerando Todos os Dados Disponíveis na Série Histórica e Previsões com o Modelo Híbrido (LSTM+GRU)

A construção de modelos preditivos, como descrito por Goodfellow, Bengio e Courville (2016), é uma tarefa central em *machine learning*. Tradicionalmente, os dados são divididos em três conjuntos: treino, validação e teste. O conjunto de treino é utilizado para ajustar os parâmetros do modelo, o conjunto de validação é empregado para otimizar os hiperparâmetros e evitar o sobreajuste, e o conjunto de teste serve para avaliar a performance final do modelo, ou seja, sua capacidade de generalização. A validação de um modelo é um passo crucial, conforme documentado por Keras (2024), além de ser amplamente discutido em livros e artigos de referência, como abordado nesta seção. Essa etapa é essencial para evitar que o modelo se ajuste excessivamente aos dados de treino, comprometendo sua capacidade de fazer previsões eficazes em dados novos.

Após a validação e o teste, uma prática comum é retreinar o modelo utilizando todos os dados disponíveis. Isso ocorre porque, se um modelo demonstrou boa capacidade de generalização nos dados de validação, ele pode se beneficiar do uso de um conjunto maior de dados para aprimorar sua precisão nas previsões futuras. De acordo com Goodfellow, Bengio e Courville (2016), TensorFlow (2024) e Keras (2024), as redes neurais recorrentes tendem a aprender de forma mais eficaz quando têm acesso a um maior volume de dados.

Neste trabalho, aplicamos a arquitetura que obteve o melhor desempenho entre as combinações estudadas para treinar toda a série histórica, visando realizar previsões a partir do último ponto existente na série, que é

setembro de 2024. A partir disso, procuramos prever os dados para o período de outubro de 2024 em diante, replicando o procedimento utilizado para prever os 12 passos à frente nos conjuntos de treino e validação. O uso de todos os dados disponíveis permite também capturar variações que podem não ter sido observadas nos subconjuntos de treino e validação.

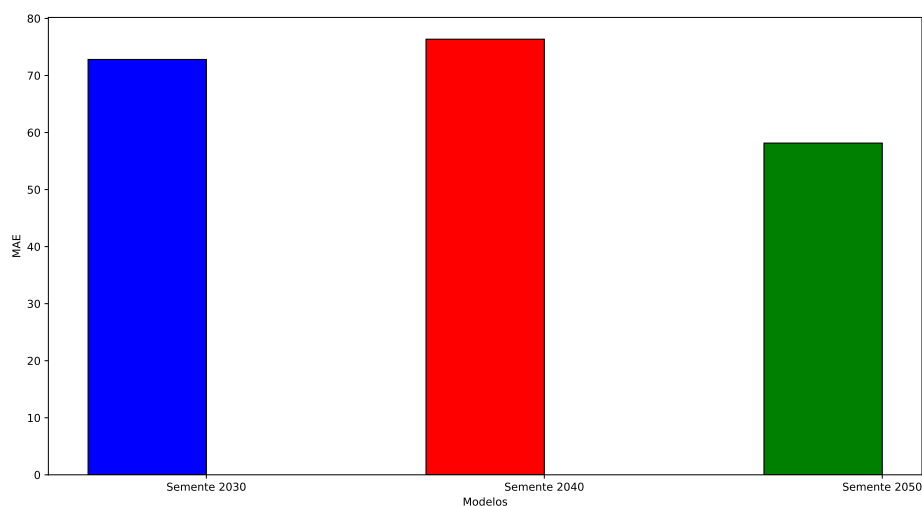
Em resumo, após a validação e o teste de um modelo, o retreinamento com dados completos é uma prática recomendada que visa maximizar o desempenho do modelo nas previsões futuras. Essa estratégia é respaldada pela lógica de que um modelo bem validado pode se beneficiar da utilização de um maior volume de dados, resultando em melhor generalização e maior precisão.

A Figura 16 mostra o desempenho do modelo híbrido (LSTM+GRU) em termos de erro absoluto médio (MAE) para as sementes 2030, 2040 e 2050. Observa-se que a semente 2040 obteve o maior erro médio, com um valor de 76, enquanto a semente 2050 apresentou o menor erro médio, com 59.

5.4.1 Métricas de Desempenho MAE Durante o Treinamento

A Figura 16 ilustra as métricas de desempenho MAE durante o treinamento. Essas métricas são fundamentais para avaliar a eficácia do modelo em prever os dados.

Figura 16 – Métricas de Desempenho MAE Durante o Treinamento



Fonte: Elaborada pelo autor, 2024.

A Tabela 4 representa o desempenho do modelo híbrido (LSTM+GRU) em termos de erro absoluto médio (MAE) nos dados de treinamento para as sementes de 2030, 2040 e 2050.

Tabela 4 – Desempenho do modelo híbrido treinado com todos os dados disponíveis.

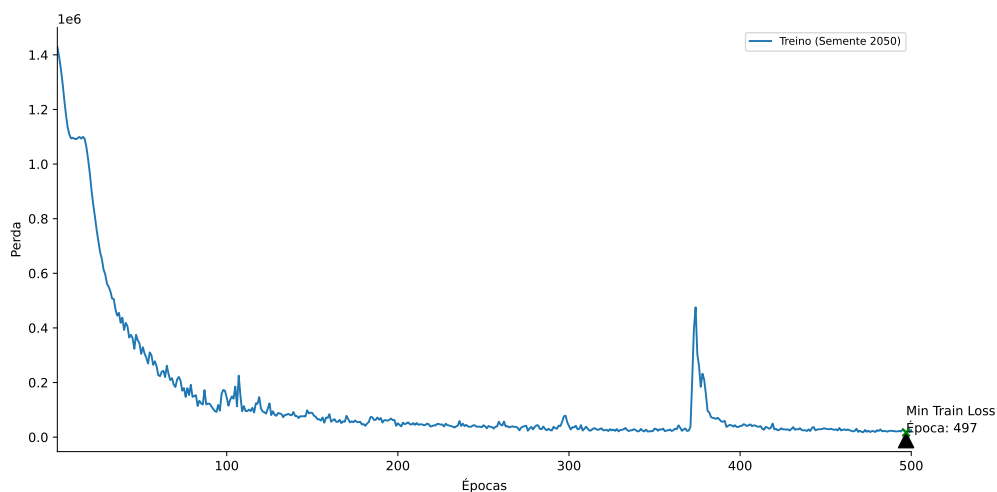
Modelo	Semente 2030	Semente 2040	Semente 2050
(LSTM+GRU)	72	76	59

Fonte: Elaborada pelo autor, 2024.

5.4.2 Histórico de Perda Durante o Treinamento

A Figura 17 apresenta o histórico de perda exclusivamente para o conjunto de treino, evidenciando que o menor erro absoluto médio foi registrado na época 497, com base nos parâmetros ajustados ao conjunto de treino completo.

Figura 17 – Histórico de perda (*Loss*) durante o treinamento.

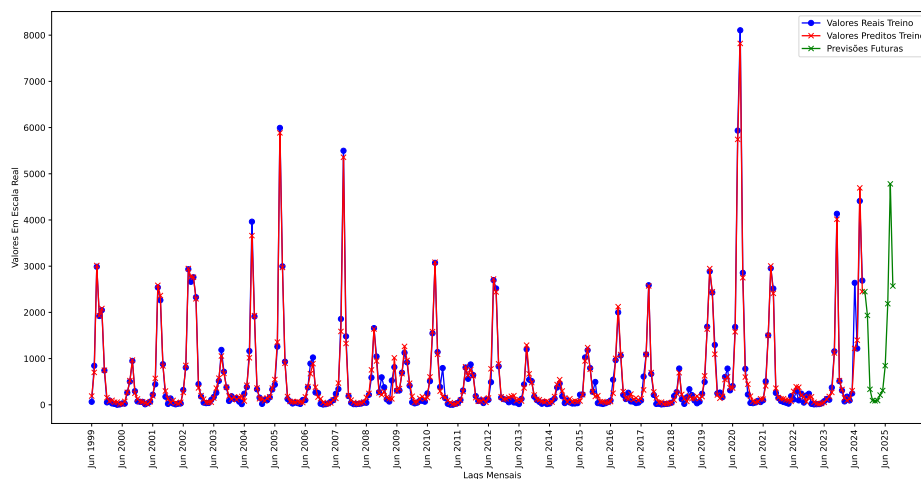


Fonte: Elaborada pelo autor, 2024.

5.4.3 Modelo Híbrido Treinado com Todos os Dados da Série Temporal

A Figura 18 exibe o conjunto de treinamento completo dos dados, que abrange junho de 1998 até setembro de 2024, bem como as previsões de 12 meses à frente. Essa visualização é essencial para compreender as tendências e flutuações dos dados ao longo do tempo.

Figura 18 – Previsões do modelo híbrido (LSTM+GRU) - Conjunto Completo.



Fonte: Elaborada pelo autor, 2024.

5.4.4 Previsões dos Meses Futuros

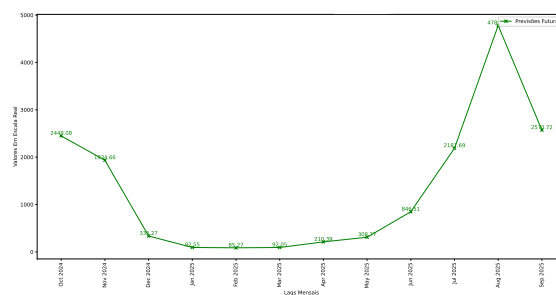
A Tabela 19 apresenta as previsões para os meses futuros da série histórica treinada, de outubro de 2024 a setembro de 2025.

Tabela 19 – Previsões dos Meses Futuros (Outubro de 2024 a Setembro de 2025).

Mês	Previsão
Outubro 2024	2449
Novembro 2024	1935
Dezembro 2024	335
Janeiro 2025	93
Fevereiro 2025	85
Março 2025	92
Abril 2025	210
Mai 2025	309
Junho 2025	847
Julho 2025	2188
Agosto 2025	4781
Setembro 2025	2570

Fonte: Elaborada pelo autor, 2024.

Figura 20 – Previsões isoladas do modelo híbrido (LSTM+GRU).



Fonte: Elaborada pelo autor, 2024.

Na Figura 20 são destacadas as previsões para os próximos 12 meses após a série histórica, permitindo uma análise detalhada de cada mês predito. As previsões foram geradas com a semente numérica fixada em 2050, garantindo a reprodutibilidade dos resultados conforme as técnicas implementadas no tutorial de Scikit-learn (2024).

Como observado, as previsões seguem a sazonalidade histórica da série temporal de focos de calor no Pantanal. A série apresenta uma queda no número de focos de calor de dezembro a maio, um padrão que é mantido nas previsões, refletindo a tendência sazonal dos dados históricos. Ao iniciar o segundo semestre, a previsão aponta para um aumento expressivo a partir de junho, com os focos de calor subindo de 847 em junho até o pico máximo de 4781 em agosto.

Esse comportamento reflete de maneira clara as flutuações sazonais observadas nos anos anteriores, confirmando que o modelo híbrido (LSTM+GRU) foi eficaz em capturar essas variações sazonais intrínsecas da série temporal. Esse desempenho destaca a capacidade dos modelos de redes neurais em aprender e reproduzir padrões temporais, especialmente aqueles relacionados à sazonalidade, um dos desafios típicos em previsões de séries temporais ambientais.

6 CONCLUSÃO

Este trabalho teve como objetivo modelar e prever a série temporal de focos de calor detectados pelo satélite *AQUA_M-T* no Bioma do Pantanal, utilizando um modelo híbrido de Redes Neurais Recorrentes (RNN), que combina as arquiteturas *Long Short-Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU). O principal objetivo foi verificar se as previsões capturaram a sazonalidade inerente dos dados, caracterizada pela repetição de picos máximos e mínimos anuais em períodos específicos. Para tanto, a metodologia adotada incluiu a preparação e o treinamento dos dados em sua escala original, sem transformações que alterassem sua natureza de contagem. A série temporal foi treinada utilizando técnicas de aprendizado de máquina, com validação cruzada realizada por meio de três sementes, o que permitiu avaliar a robustez do modelo em diferentes cenários.

Os resultados indicaram que o modelo híbrido de LSTM e GRU foi eficaz em aprender os padrões complexos dos dados e capturar a sazonalidade da série temporal, apresentando uma redução consistente do erro absoluto médio (MAE) durante as épocas de treinamento. Esse desempenho reforça a adequação da abordagem híbrida para séries temporais com alta sazonalidade, demonstrando superioridade em relação ao uso isolado de cada arquitetura. Além de alcançar os objetivos propostos, o estudo contribuiu para o campo do aprendizado de máquina aplicado ao monitoramento ambiental, ao detalhar a estruturação do modelo, a configuração dos parâmetros e a adaptação dos pesos para dados de contagem. Essa descrição pode servir como referência para futuras pesquisas na área de previsão de séries temporais ambientais.

Contudo, algumas limitações foram identificadas, como o aumento do custo computacional devido à combinação dos modelos e a necessidade de ajustes mais refinados de parâmetros, como taxa de aprendizado e número de neurônios.

Para estudos futuros, recomenda-se testar outras taxas de aprendizado, explorar diferentes algoritmos de otimização além do *Adam*, e avaliar o impacto de modelos isolados com ajustes mais finos. Essas ações visam aprimorar ainda mais a precisão das previsões.

Em suma, este estudo demonstrou que a combinação de redes neurais recorrentes é uma estratégia eficaz para a previsão de séries temporais com alta sazonalidade, contribuindo para o desenvolvimento de metodologias de monitoramento e previsão de eventos ambientais críticos.

REFERÊNCIAS

- ABURASS, S.; DORGHAM, O.; SHAQSI, J. A. **A hybrid machine learning model for classifying gene mutations in cancer using LSTM, BiLSTM, CNN, GRU, and GloVe**. 2024. Acesso em: 30 out. 2024. Disponível em: <<https://arxiv.org/abs/2307.14361>>. Citado na página 28.
- AHMAD, R. A.; AZHAR, M.; SATTER, H. **An image captioning algorithm based on the hybrid deep learning technique (CNN+GRU)**. 2024. Acesso em: 30 out. 2024. Disponível em: <<https://arxiv.org/abs/2403.06789>>. Citado na página 21.
- BENIDIS, K.; RANGAPURAM, S. S.; FLUNKERT, V.; WANG, Y.; MADDIX, D.; TURKMEN, C.; GASTHAUS, J.; BOHLKE-SCHNEIDER, M.; SALINAS, D.; STELLA, L.; AUBET, F.-X.; CALLOT, L.; JANUSCHOWSKI, T. Deep learning for time series forecasting: tutorial and literature survey. **ACM Computing Surveys**, Association for Computing Machinery (ACM), v. 55, n. 6, p. 1–36, 2022. ISSN 1557-7341. Acesso em: 30 out. 2024. Disponível em: <<http://dx.doi.org/10.1145/3533382>>. Citado nas páginas 10 e 27.
- BOX, G. E. P.; JENKIS, G. M.; REINSEL, G. C.; LJUNG, G. M. **Time Series Analysis: Forecasting and Control**. Wiley, 2015. 1-712 p. Disponível em: <<https://www.researchgate.net/publication/>>. Citado na página 10.
- BROCKWELL, P. J.; DAVIS, R. A. **Introduction to time series and forecasting**. New York: Springer, 2002. 1-436 p. Citado na página 10.
- CHENG, L.; PANDEY, A.; XU, B.; DELBRUCK, T.; LIU, S. C. Dynamic gated recurrent neural network for compute-efficient speech enhancement. **Proceedings of Interspeech 2024**, p. 1–5, 2024. Disponível em: <<http://dx.doi.org/10.21437/Interspeech.2024-958>>. Acesso em: 01 nov. 2024. Citado nas páginas 14 e 15.
- CHUNG, J.; GULCEHRE, C.; CHO, K.; BENGIO, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. **CoRR**, abs/1412.3555, p. 1–9, 2014. Disponível em: <<http://arxiv.org/abs/1412.3555>>. Acesso em: 01 nov. 2024. Citado nas páginas 10 e 14.
- CLEVELAND, R. B.; CLEVELAND, W. S.; MCRAE, J. E.; TERPENNING, I. Stl: A seasonal-trend decomposition procedure based on loess. **Journal of Official Statistics**, v. 6, n. 1, p. 3–32, 1990. Citado nas páginas 18 e 19.
- GAO, S.; HUANG, Y.; ZHANG, S.; HAN, J.; WANG, G.; ZHANG, M.; LIN, Q. Short-term runoff prediction with gru and lstm networks without requiring time step optimization during sample generation. **Journal of Hydrology**, v. 589, p. 125–188, 2020. ISSN 0022-1694. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S002216942030648X>>. Citado na página 21.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, 2016. 166-173 p. Disponível em: <<https://www.deeplearningbook.org/>>. Acesso em: 01 nov. 2024. Citado nas páginas 15, 16, 20 e 31.
- GRAVES, A.; MOHAMED, A.; HINTON, G. Speech recognition with deep recurrent neural networks. **arXiv preprint**, abs/1303.5778, p. 1–5, 2013. Disponível em: <<https://doi.org/10.48550/arXiv.1303.5778>>. Acesso em: 01 nov. 2024. Submetido em: 22 mar. 2013. Citado na página 12.
- GREFF, K.; SRIVASTAVA, R. K.; KOUTNÍK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. Lstm: A search space odyssey. **IEEE Transactions on Neural Networks and Learning Systems**, v. 28, n. 10, p. 2222–2232, 2017. Disponível em: <<https://doi.org/10.48550/arxiv.1503.04069>>. Acesso em: 01 nov. 2024. Citado nas páginas 12, 13 e 14.
- GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**. Sebastopol: O'Reilly Media, Inc., 2017. 1-707 p. Acesso em: 2 nov. 2024. Disponível em: <<https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>>. Citado na página 20.

- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997. Disponível em: <<https://doi.org/10.1162/neco.1997.9.8.1735>>. Acesso em: 01 nov. 2024. Citado na página 10.
- INPE. **Situação atual das queimadas no Brasil**. 2024. Acesso em: 12 ago. 2024. Disponível em: <http://terrabrasilis.dpi.inpe.br/queimadas/situacao-atual/estatisticas/estatisticas_estados/>. Citado nas páginas 20 e 23.
- JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An introduction to statistical learning**. New York: Springer, 2013. 1-424 p. ISBN 978-1-4614-7138-7. Disponível em: <<https://www.springer.com/gp/book/9781461471387>>. Citado na página 10.
- KERAS. **Activation functions**. 2024. Acesso em: 29 out. 2024. Disponível em: <<https://keras.io/api/layers/activations/>>. Citado nas páginas 16, 18, 20, 22, 23 e 31.
- KINGMA, D. P.; BA, J. L. **Adam: A method for stochastic optimization**. 2017. 1-15 p. Disponível em: <<https://arxiv.org/abs/1412.6980>>. Acesso em: 01 nov. 2024. Submetido em: 22 dez. 2014 (v1), última revisão em: 30 jan. 2017 (v9). Citado nas páginas 16, 17 e 18.
- LINDEMANN, B.; MÜLLER, T.; VIETZ, H.; JAZDI, N.; WEYRICH, M. A survey on long short-term memory networks for time series prediction. **Procedia CIRP**, v. 99, p. 650–655, 2021. ISSN 2212-8271. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 jul. 2020. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2212827121003796>>. Citado na página 21.
- Ministério do Meio Ambiente e Mudança do Clima. **Plano de ação para prevenção e controle do desmatamento e das queimadas no Pantanal**. 2024. Acesso em: 29 out. 2024. Disponível em: <https://www.gov.br/participamaisbrasil/plano-de-acao-para-prevencao-e-controle-do-desmatamento-e-das-queimadas-no-pantanal-?fbclid=PAZXh0bgNhZW0CMTEAAaaFE2cq3A-fBNvRK14g4HKtdW3Uk4gTVgeLTJENV_lGovqffXyMuQGtq_c_aem_LZDyshDfWbMBiyELC9slWw>. Citado nas páginas 25 e 26.
- MOUSA, R.; KHEZLI, M.; AZADI, M.; NIKOOAFARD, V.; HESARAKI, S. **Classifying objects in 3D point clouds using recurrent neural network: a GRU LSTM hybrid approach**. 2024. Acesso em: 02 nov. 2024. Disponível em: <<https://arxiv.org/abs/2403.05950>>. Citado na página 21.
- PYTHON. **Python programming language**. 2024. <<https://www.python.org/>>. Acesso em: 05 ago. 2024. Citado nas páginas 19, 20 e 22.
- RUDER, S. **An overview of gradient descent optimization algorithms**. 2017. 1-14 p. Disponível em: <<https://arxiv.org/abs/1609.04747>>. Acesso em: 01 nov. 2024. Citado na página 17.
- SAINATH, T. N.; KINGSBURY, B.; MOHAMED, A.; DAHL, G. E.; SAON, G.; SOLTAU, H.; BERAN, T.; ARAVKIN, A. Y.; RAMABHADHAN, B. Improvements to deep convolutional neural networks for lvcscr. **CoRR**, abs/1309.1501, p. 1–7, 2013. Disponível em: <<http://arxiv.org/abs/1309.1501>>. Acesso em: 01 nov. 2024. Citado na página 18.
- SCIKIT-LEARN. **Scikit-learn: machine learning in python**. 2024. Acesso em: 05 ago. 2024. Disponível em: <<https://scikit-learn.org/stable/index.html>>. Citado nas páginas 20, 21 e 34.
- SEIA. **Situação atual das queimadas florestais**. 2024. Acesso em: 04 set. 2024. Disponível em: <<http://www.seia.ba.gov.br/monitoramento-ambiental/focos-de-calor>>. Citado na página 10.
- STATSMODELS. **Seasonal-Trend decomposition using LOESS (STL)**. 2024. Acesso em: 02 nov. 2024. Disponível em: <https://www.statsmodels.org/dev/examples/notebooks/generated/stl_decomposition.html>. Citado nas páginas 18 e 19.
- TAVARES, R.; OLINDA, R. **Neural Networks with LSTM and GRU in Modeling Active Fires in the Amazon**. 2024. Acesso em: 01 nov. 2024. Disponível em: <<https://arxiv.org/abs/2409.02681>>. Citado nas páginas 10, 21 e 27.

TENSORFLOW. **TensorFlow: a platform for machine learning**. 2024. Acesso em: 05 ago. 2024. Disponível em: <https://www.tensorflow.org/tutorials/structured_data/time_series?hl=pt-br>. Citado nas páginas 20, 22, 23 e 31.

WANG, Y.; WANG, W.; ZANG, H.; XU, D. Is the lstm model better than rnn for flood forecasting tasks? a case study of huayuankou station and loude station in the lower yellow river basin. **Water**, v. 15, n. 22, p. 1–26, 2023. Disponível em: <<https://www.mdpi.com/2073-4441/15/22/3928>>. Acesso em: 01 nov. 2024. Citado na página 21.