

ANÁLISE COMPARATIVA DE MODELOS MLP E LSTM EM PROCESSO DE PREDIÇÃO DE CASOS DE DENGUE NO BRASIL

¹BELLA, Matheus Henrique
²TORRES, Giovanni Micheletti
³CASTRO, Marcelo George de
⁴NEGRETTO, Diego

Centro Universitário Hermínio Ometto – FHO, Araras – SP, Brasil

Resumo

Este trabalho tem como objetivo desenvolver um modelo de análise preditiva para casos de dengue, utilizando Redes Neurais Artificiais (RNA) do tipo Multilayer Perceptron (MLP) e Long Short-Term Memory (LSTM). A dengue tem apresentado crescimento expressivo no Brasil, sendo influenciada por diversos fatores, principalmente variáveis climáticas como temperatura, umidade e precipitação. Nesse contexto, a modelagem preditiva baseada em técnicas de aprendizado de máquina surge como uma alternativa promissora para análise de dados históricos e identificação de padrões que favoreçam a ocorrência da doença. O estudo envolveu a coleta, limpeza e transformação de dados epidemiológicos e climáticos, abrangendo o período de 2014 a 2023, com fontes como INMET e DATASUS. Durante o desenvolvimento, foram aplicadas técnicas de pré-processamento, como normalização, codificação de variáveis categóricas e padronização dos dados, visando garantir a qualidade do treinamento da rede. O modelo foi treinado utilizando o algoritmo Backpropagation, com monitoramento por métricas como Erro Quadrático Médio (MSE) e Erro Absoluto Médio (MAE), além de técnicas de regularização para evitar o overfitting. Os experimentos mostraram que a rede neural MLP possui maior precisão e eficiência. O modelo alcançou um Erro Quadrático Médio (MSE) de 0,00126 e um Erro Absoluto Médio (MAE) de 0,00525, superando o LSTM em 24,5% e 20,5%, respectivamente. Além disso, o MLP foi aproximadamente 31,4% mais rápido em seu tempo de execução. A comparação entre as arquiteturas permitiu gerar previsões para o ano de 2024 em todos os estados brasileiros. Como resultado, o trabalho apresenta uma ferramenta de apoio à tomada de decisão para órgãos públicos de saúde, possibilitando a criação de estratégias preventivas mais eficazes no combate à doença.

Palavras chave: Redes Neurais Artificiais (RNA), Multilayer Perceptron (MLP), Monitoramento por métricas.

¹FHO|UNIARARAS. Aluno do Curso de Sistemas de Informação, Matheus Henrique Bella, matheus.bella07@alunos.fho.edu.br

²FHO|UNIARARAS. Aluno do Curso de Sistemas de Informação, Giovanni Micheletti Torres, giovanni.micheletti@alunos.fho.edu.br

³FHO|UNIARARAS. Professor do Curso de Sistemas de Informação, professor. Me. Marcelo George de Castro, marcelocastro@fho.edu.br

⁴FHO|UNIARARAS. Professor do Curso de Sistemas de Informação, professor. Me. Diego Negretto, diegonegretto@fho.edu.br

1 Introdução

1.1 Contextualização

O uso da análise de dados no combate a doenças como a dengue tem se mostrado uma ferramenta valiosa para prever casos e auxiliar os governos na preparação e realocação de recursos (CASTRO et al., 2022). Com o avanço das técnicas de aprendizado de máquina e a disponibilidade de dados históricos, a modelagem preditiva torna-se uma alternativa viável para antecipar cenários futuros (PHILIPS, 2020).

A dengue, em particular, destaca-se por ser uma das principais ameaças à saúde pública no Brasil, apresentando crescimento sem precedentes nos últimos anos. Um dos principais fatores que contribuem para os surtos de dengue no país é a mudança climática, que agrava ainda mais o cenário e dificulta o controle da doença (SIQUEIRA et al., 2018).

Os fatores climáticos desempenham um papel fundamental na proliferação do mosquito *Aedes aegypti*, influenciando diretamente a dinâmica de transmissão da doença. Entre os principais elementos estão a precipitação, a temperatura e a umidade. O aumento da precipitação e da umidade proporciona ambientes propícios para a formação de criadouros, enquanto o aumento da temperatura acelera o ciclo de vida do mosquito, reduzindo o tempo necessário para que os ovos se desenvolvam até a fase adulta (CEARÁ, 2024).

Diante do crescimento alarmante da dengue, a criação de ferramentas eficazes para combatê-la tornou-se urgente (MATTIOLI; ANDRADE; ESTEVEZ, 2017). Nesse contexto, este trabalho propõe a utilização de técnicas de análise de dados e aprendizado de máquina, com foco em redes neurais, para desenvolver modelos preditivos capazes de antecipar surtos de dengue.

Os modelos serão alimentados por múltiplas fontes, como registros históricos de casos e de mortes por dengue e condições climáticas que são temperatura e precipitação (INMET). A combinação dessas informações permitirá identificar tendências que possam indicar a probabilidade de novos surtos.

1.2 Tema de Pesquisa

Este estudo propõe uma análise preditiva dos casos de dengue, utilizando redes neurais do tipo MLP (*Multilayer Perceptron*) e LSTM (*Long Short-Term Memory*). Essa arquitetura, aplicada em previsão de séries temporais e identificação de padrões complexos (FERNANDES; PORTUGAL; NAVAUX, 1995), permite gerar previsões mais precisas, considerando a influência de fatores climáticos na evolução da doença.

1.3 Motivações e Justificativas

A escolha deste tema deve-se ao aumento crescente dos casos de dengue em todo o país, tornando-se uma grande preocupação. Diante disso, surge a necessidade de prever surtos da doença, possibilitando que as instituições governamentais tomem medidas antecipadas e desenvolvam estratégias mais eficazes. Isso permite otimizar a alocação de recursos e reduzir o impacto da dengue na população.

Atualmente, a maioria dos estudos sobre previsões epidemiológicas utiliza métodos estatísticos. Este trabalho investiga o uso de redes neurais não-lineares para ganho de precisão.

Além disso, este estudo busca contribuir para o aperfeiçoamento das técnicas de análise de dados aplicadas à saúde pública, demonstrando a eficiência das redes neurais do tipo MLP e LSTM.

1.4 Objetivos

1.4.1 Objetivo Geral

Desenvolver e avaliar o modelo de análise preditiva para casos de dengue com redes neurais artificiais: *Multilayer Perceptron* e *Long Short-Term Memory*, considerando variáveis que podem influenciar a doença e aplicando métricas para essa validação.

1.4.2 Objetivos Específicos

- Coletar e organizar bases de dados sobre a incidência de dengue no período determinado;
- Aplicar modelos de aprendizado de máquina, com ênfase em redes neurais do tipo MLP (*Multilayer Perceptron*) e LSTM (*Long Short-Term Memory*);
- Medir a eficiência dos modelos (MLP e LSTM) em séries temporais para a predição dos casos de dengue no Brasil a partir de um estudo sobre séries temporais;
- Medir o tempo de execução (TE) e a assertividades do modelo preditivo por meio de métricas padronizadas, como Erro Quadrático Médio (MSE) e Erro Absoluto Médio (MAE);

2 Revisão Bibliográfica

2.1 Fundamentação Teórica e Técnica

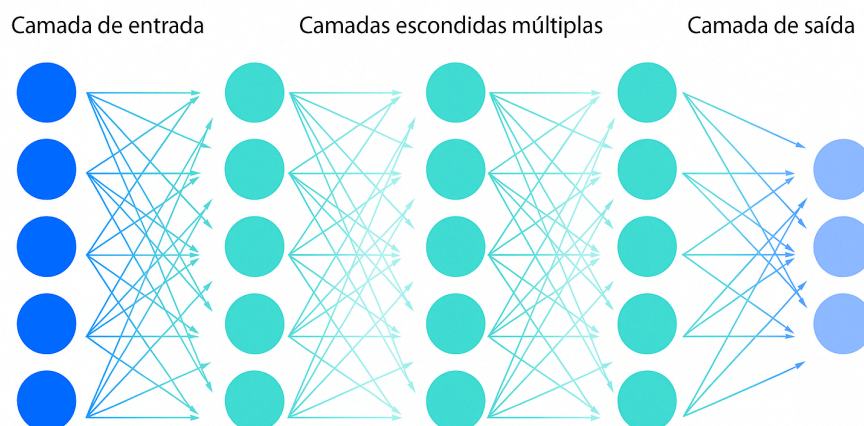
As Redes Neurais Artificiais (RNA) são modelos computacionais inspirados no funcionamento dos neurônios biológicos e amplamente utilizadas em tarefas de reconhecimento de padrões, previsão de séries temporais e classificação de dados. Esses modelos são capazes de aprender padrões complexos a partir de dados (GRÜBLER, 2018) e podem ser aplicados na análise epidemiológica da dengue.

2.1.1 Multilayer Perceptron (MLP)

A arquitetura MLP é uma das mais utilizadas para problemas de classificação e regressão. Conforme mostrado na Figura 1, ela consiste de três principais camadas: a camada de entrada, que recebe os dados brutos correspondentes às variáveis climáticas, temporais, geográficas e epidemiológicas, as camadas ocultas, responsáveis por realizar transformações matemáticas sobre os dados utilizando e funções de ativação para extrair padrões complexos e a camada de saída, que gera as previsões (GRÜBLER, 2018), podendo indicar o nível de risco de incidência de dengue.

Figura 1: Arquitetura de um modelo MLP com três camadas ocultas

Rede Neural Profunda



Fonte - IBM, 2025 (Adaptado)

2.1.2 Long Short-Term Memory (LSTM)

A arquitetura da LSTM consiste em uma única camada de unidade LSTM seguida por uma camada densa. A camada LSTM possui 4 unidades e recebe como entrada um tensor 3D com a forma (*batch_size*, *look_back*, *n_features*), onde *look_back* é o número de passos de tempo retroativos considerados e *n_features* é o número de variáveis de entrada nos dados. A camada densa possui uma única unidade e produz um tensor 2D com a forma (*batch_size*, 1) (MAJEED, 2025).

As redes *Long Short-Term Memory* (LSTM) são uma extensão das redes neurais recorrentes (RNN) projetadas para lidar com o problema do desvanecimento e explosão do gradiente, permitindo a modelagem de dependências de longo prazo em séries temporais. De acordo com Evsukoff (2020), as LSTM foram introduzidas por Hochreiter e Schmidhuber e utilizam células de memória que contém o vetor de estados internos da rede, e um conjunto de portas que controlam o fluxo de informação dentro do neurônio.

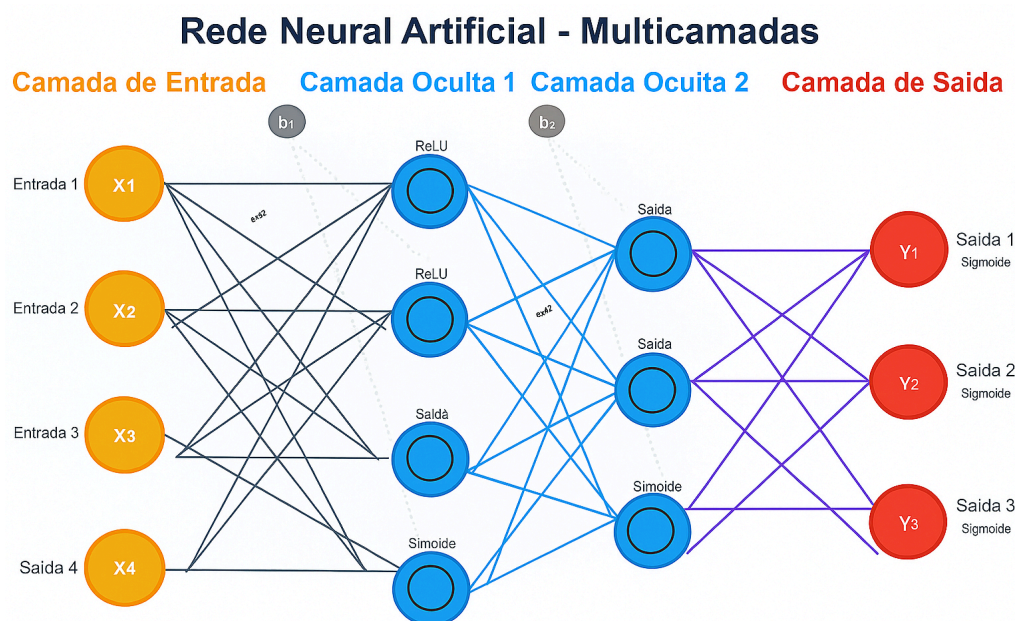
Essas portas incluem:

- Porta de esquecimento: controla a proporção do estado anterior a ser mantida;
- Porta de entrada: decide quais valores novos devem ser armazenados;
- Porta de saída: determina quais partes do estado serão usadas como saída.

Conforme ilustrado na Figura 2, a arquitetura da LSTM permite que o erro seja retropropagado diretamente através do tempo (BPTT), reduzindo problemas de gradiente nulo ou infinito, possibilitando o treinamento estável em séries longas. Segundo Mghazli utilizado como referência, as LSTM se destacam em aplicações como previsão de séries temporais financeiras, reconhecimento de fala e tradução automática (MGHAZLI, 2021).

A topologia da LSTM consiste em camadas de unidades LSTM conectadas a uma camada de saída, geralmente utilizando funções de ativação como ReLU ou *sigmóide*, dependendo do problema. O treinamento é realizado utilizando algoritmos de otimização como o Adam, com função de perda baseada no erro médio quadrático (MSE), conforme recomendado por Mghazli (2021).

Figura 2: Arquitetura de um modelo LSTM



Fonte - Autor, 2025

2.1.3 Funções de ativação

As funções de ativação introduzem não linearidade ao modelo, permitindo a aprendizagem de padrões complexos (LEITE, 2018). Uma das mais utilizadas é a ReLU (*Rectified Linear Unit*), definida pela seguinte função:

$$f(x) = \max(0, x)$$

Essa função retorna zero para valores negativos e mantém valores positivos inalterados, promovendo uma convergência mais rápida durante o treinamento. A ReLU é amplamente usada por sua simplicidade computacional e por reduzir o problema do desaparecimento do gradiente, comum em funções *sigmóides*.

2.1.4 Métricas de Erro

A avaliação da qualidade das previsões realizadas pelo modelo é feita por meio de métricas de erro. As duas mais comuns em regressão são:

2.1.4.1 Erro Absoluto Médio (MAE - *Mean Absolute Error*)

Mede a média dos valores absolutos das diferenças entre os valores reais e previstos (FERNANDES; PORTUGAL; NAVAUX, 1995). Sua fórmula é:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.1)$$

n = número total de observações,

\hat{y}_i = valor previsto,

y_i = valor real.

O MSE é caracterizado por ser menos sensível a outliers, interpretação direta.

2.1.4.2 Erro Quadrático Médio (MSE - *Mean Squared Error*)

Mede a média dos quadrados das diferenças entre os valores previstos e os valores reais. Sua fórmula é:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.2)$$

y_i = valor real,

\hat{y}_i = valor previsto,

n = número total de observações.

Essa métrica penaliza erros grandes de forma mais severa (JÚNIOR, 2021).

A característica do MAE é menos sensível a outliers, interpretação direta.

2.1.5 Cálculo Matemático do Tempo de Execução

- O Tempo de Execução em redes neurais refere-se ao tempo necessário para que a rede processe uma entrada ou (lote de entradas) e gere a saída durante a fase de inferência, ou seja, após o treinamento do modelo, quando ele é utilizado para realizar previsões em dados novos.

$$T_{exec} = t_{end} - t_{start} \quad (2.3)$$

t_{start} = instante em que a inferência inicia

t_{end} = instante em que a inferência finaliza (GERON, 2019).

2.1.6 Estratégias de Treinamento

Ao treinar redes neurais, é crucial empregar métodos que previnam o *overfitting*, situação onde o modelo memoriza os dados de treino e falha ao lidar com dados inéditos. Há duas formas eficazes de mitigar isso:

- *Early Stopping*: interrompe automaticamente o treinamento quando a performance do modelo nos dados de validação para de melhorar após um certo número de épocas consecutivas (*patience*). Isso evita que o modelo continue aprendendo ruídos dos dados e degrade sua capacidade de generalização;
- *ModelCheckpoint*: permite salvar automaticamente o modelo durante o treinamento, geralmente armazenando a versão que obteve o melhor desempenho em um conjunto de validação. Dessa forma, mesmo que o modelo continue treinando e sofra degradação, a melhor versão já estará salva (Silva, 2019).

A relação entre fatores climáticos e a propagação da dengue é um ponto essencial do estudo. A precipitação influencia diretamente na proliferação do *Aedes aegypti*, criando criadouros para o mosquito. A temperatura média afeta o ciclo de vida do mosquito, acelerando sua reprodução e a transmissão do vírus. Já a umidade relativa desempenha um papel crucial na sobrevivência do mosquito, aumentando o risco de transmissão da doença (LEITE, 2023).

A utilização de redes neurais artificiais para analisar a influência das variáveis climáticas na incidência de dengue proporciona uma abordagem promissora para previsão e monitoramento epidemiológico. A implementação de métodos adequados de treinamento e avaliação do modelo é essencial para garantir previsões precisas e eficientes, contribuindo para a formulação de estratégias de combate à doença (FERNANDES; PORTUGAL; NAVAUX, 1995).

2.2 Trabalhos Relacionados

Pode-se citar alguns exemplos de trabalhos relacionados para o desenvolvimento de aplicação para redes neurais:

O trabalho de Silva (2019) abordava a teoria das redes neurais e sobre o treinamento da rede neural MLP (*Multilayer Perceptron*) e CNN (*Convolutional Neural Network*).

O estudo de Leite (2023) analisou a relação entre clima e casos de dengue em Uberlândia, mostrando que temperaturas mínimas acima de 18 °C, médias entre 26 °C e 28 °C, e umidade acima de 70% favoreceram o aumento dos casos. Já temperaturas máximas acima de 30 °C reduziram os registros. Além disso, cada região da cidade apresentou padrões climáticos distintos que influenciaram a incidência da doença. Os resultados podem apoiar ações para controlar surtos.

A dissertação de Reis (2023) propõe o uso de uma rede neural convolucional (CNN), especificamente a YOLOv8, para identificar piscinas — potenciais criadouros do mosquito da dengue — em imagens de sensoriamento remoto. Após preparar e treinar o modelo com dados anotados, o desempenho foi avaliado, destacando a eficácia da abordagem na detecção automática desses focos, contribuindo para o combate à doença.

3 Metodologia

Todos os códigos desenvolvidos durante este trabalho foram versionados e disponibilizados publicamente no GitHub, garantindo rastreabilidade e reprodutibilidade.

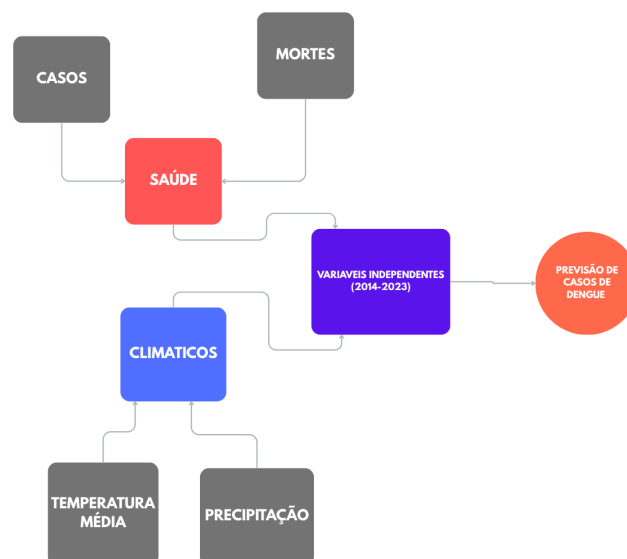
Repositório: https://github.com/GiovanniTT/RedeNeural_TCC

3.1 Procedimentos Metodológicos

3.1.1 Coleta de dados

Para o desenvolvimento deste projeto, foram definidas quatro variáveis: casos, mortes, temperatura média e precipitação. Conforme pode ser observado na Figura 3, todas essas variáveis foram utilizadas com o objetivo de prever a incidência de casos de dengue. O recorte de dados refere-se ao período de 2014 a 2023 e abrange todos os estados brasileiros.

Figura 3: Ilustração das variáveis dos modelos de predição de dengue



Fonte - Autor, 2025

A coleta de dados foi realizada a partir de fontes oficiais e confiáveis: informações climáticas (precipitação e temperatura média) do INMET ([Instituto Nacional de Meteorologia](#)), e registros de casos e óbitos relacionados à dengue obtidos do DATASUS ([Departamento de Informática do SUS](#)).

Para obter os dados de casos e mortes, deve-se acessar o TabNet, um painel do DATASUS onde se encontram todos os dados relacionados à dengue. O apêndice A é um tutorial de como utilizar o TABNET, por meio dele é possível obter as informações.

A configuração utilizada para obter os dados de casos foi a seguinte: na linha, "Mês de Notificação"; e na coluna, "UF de Residência". Para o período, foram realizadas diversas consultas, uma para cada ano. Já para os dados de óbitos, manteve-se a configuração anterior e selecionou-se também, na aba "Evolução", a opção "Óbito pelo agravo notificado".

Foram obtidos os dados de precipitação e temperatura por meio do Instituto Nacional de Meteorologia (INMET), nele é apresentado os dados, em CSVs, são segmentados por estações meteorológicas e os dados são apresentados por hora, o apêndice B apresenta-se um tutorial de como obter os dados.

Após a coleta, todos os dados foram organizados em diretórios específicos. Os dados climáticos foram separados em pastas por ano, por exemplo: 2014 e 2015, cada uma contendo os respectivos arquivos daquele período.

Para os dados de casos e óbitos, adotou-se uma estrutura semelhante, foi criada uma pasta denominada "dados_casos_mortes", na qual os arquivos seguem um padrão de nomenclatura. Os arquivos nomeados apenas com o ano por exemplo: 2016 e 2017 correspondem aos dados de casos, enquanto aqueles que incluem a letra "d" após o ano por exemplo: 2016d e 2017d referem-se aos óbitos.

Essa padronização é fundamental, pois foram desenvolvidos códigos para automatizar as rotinas de limpeza, tratamento, extração das informações de interesse e por fim, a criação do Banco de Dados.

3.1.2 Tratamento e Limpeza dos Dados

Os dados coletados foram submetidos a um processo de ETL (*Extract, Transform, Load*), que incluiu:

- Remoção de inconsistências, como valores ausentes e duplicados;
- Normalização das unidades de medida.

Durante a etapa de transformação dos dados para a rede neural, foi aplicada a técnica de normalização, com o objetivo de garantir que todas as variáveis de entrada e saída estivessem em uma escala numérica adequada para o aprendizado da rede (LUCAS, 2024).

- **Para as variáveis de entrada (*features X*):** As variáveis de entrada, também chamadas de *features X*, são os dados utilizados pela rede neural como insumos para realizar as previsões (BELTA.IO, 2025).
 - Utilizada a técnica de padronização por meio do *StandardScaler*, da biblioteca *Scikit-learn*;

- As variáveis de entrada incluem características como o ano, mês, mortes de dengue, estado e variáveis climáticas (temperatura e precipitação), foram padronizadas utilizando o *StandardScaler*;
- A aplicação do *StandardScaler* transformou os dados de entrada para que cada *feature* tivesse média zero (0) e desvio padrão igual a um (1). Essa padronização foi fundamental para evitar que variáveis com escalas maiores tivessem mais influência no processo de ajuste dos pesos da rede.

$$X = \frac{X - \mu}{\sigma} \quad (3.1)$$

- X = valor original da variável,
- μ = média da variável
- σ = desvio padrão da variável.

A tabela 1 apresenta uma amostra dos dados normalizados utilizados no treinamento do modelo preditivo de dengue, contendo registros referentes ao período de 2014 a 2023. A estrutura dos dados inclui variáveis como o ano, codificação dos estados, e codificação dos meses, todas pré-processadas e normalizadas para otimizar o desempenho da rede neural.

Tabela 1 – Amostra dos dados normalizados (arredondados em duas casas decimais) de variáveis de entrada (*features X*), utilizados no treinamento do modelo preditivo de dengue.

Ano	Mortes	Temperatura	Precipitação	Mês_Abril	Estado_AM
-1.57...	- 0.27...	0.91...	-0.14...	-0.30...	-0.20...
-1.57...	- 0.27...	0.91...	1.01...	-0.30...	5.10...
-1.57...	- 0.27...	0.33...	0.81...	-0.30...	-0.20...
-1.57...	- 0.27...	1.02...	-0.49...	-0.30...	-0.20...

Fonte - Autor, 2025

- **Para as variáveis de saída (*target y*):** São os valores que o modelo deve aprender a prever em um problema supervisionado (H20.AI, 2025).
 - Aplicação da normalização Min-Max, por meio do *MinMaxScaler*, também da *Scikit-learn*;
 - As variáveis de saída da rede (neste caso, o número de casos) foram normalizadas para um intervalo entre 0 e 1 utilizando o *MinMaxScaler*;
 - Essa escolha foi necessária porque a camada de saída da rede neural utiliza uma ativação linear, permitindo a geração de valores em qualquer escala. A

normalização para um intervalo fixo facilita o treinamento e melhora o desempenho da função de perda (MSE – *Mean Squared Error*).

$$Y = \frac{y - y_{min}}{y_{max} - y_{min}} \quad (3.2)$$

- y = valor original da variável,
- y_{min} e y_{max} = menor e maior valor da variável na base de dados de treinamento.

A tabela 2 apresenta uma amostra dos dados normalizados de saída utilizados no treinamento do modelo preditivo de dengue, abrangendo registros de casos e mortes referentes ao período de 2014 a 2023. Os valores foram pré-processados e normalizados na faixa de 0 a 1 utilizando a técnica *Min-Max Scaling*, visando facilitar o aprendizado do modelo e otimizar o desempenho da rede neural durante o processo de treinamento.

Tabela 2 – Amostra dos dados normalizados de variáveis de saída (*target y*) utilizados para o treinamento do modelo preditivo de dengue

Casos Normalizados
0.000018527181691
0.0
0.000004631795423
0.0

Fonte - Autor, 2025

Conforme ilustrado na Tabela 3, apresenta-se a estrutura do banco de dados utilizado, contendo registros de casos de dengue referentes ao período de 2014 a 2023.

Tabela 3 – Amostra dos dados do modelo preditivo da dengue

Id	Ano	Mês	Estado	Casos	Mortes	Temperatura (°C)	Precipitação (mm)
433	2023	Janeiro	BA	3840	20	27.0	3197.4
434	2023	Janeiro	RJ	1606	15	26.0	7008.8
435	2023	Janeiro	RS	96	3	27.5	3061.0
436	2023	Janeiro	SP	15845	26554	23.0	9182.6
437	2023	Fevereiro	BA	5067	16	28.0	1323.0
438	2023	Fevereiro	RJ	3178	7	27.5	5057.2

Fonte - Autor, 2025

3.1.3 Ferramentas e Tecnologia

O estudo faz uso das seguintes ferramentas:

- **Linguagens e bibliotecas:** Python (3.9.7) com bibliotecas como Pandas (2.2.3), Numpy (1.26.4), Matplotlib (3.8.4), Scikit-Learn (1.3.2) para análise de dados; TensorFlow (2.13.0) e Keras (2.13.1) para desenvolvimento e treinamento do modelo preditivo;
- **Funções de ativação:** ReLU para camadas ocultas (evita gradientes nulos) e função linear para a camada de saída (prevê valores contínuos);
- **Métricas de erro:** Erro Quadrático Médio (MSE) para penalizar grandes desvios e Erro Absoluto Médio (MAE) para tratar erros de forma uniforme;
- **Armazenamento:** MySQL (8.0), para consulta e manipulação de dados estruturados.

3.1.4 Análise de Dados

A análise dos dados conduzida em três etapas principais:

- **Análise exploratória:** Visa compreender os dados, identificar padrões e relações entre variáveis;
- **Análise preditiva:** Foca na construção de modelos capazes de prever casos por dengue com base em dados históricos (2014 a 2023), utilizando duas abordagens distintas de redes neurais: *Multilayer Perceptron* (MLP) e *Long Short-Term Memory* (LSTM).

3.2 Validação e Confiabilidade

Para garantir a confiabilidade e validade dos resultados, as seguintes práticas foram adotadas:

- Os dados foram obtidos a partir de fontes confiáveis, como órgãos oficiais de saúde;

- Utilização de *Early Stopping* para evitar *overfitting* durante o treinamento;
- É prática comum reservar cerca de 20% dos dados para o conjunto de teste e 80% para treino com o objetivo de avaliar a capacidade de generalização do modelo (GERON, 2019).
- Para garantir que essa divisão nós aplicamos a validação cruzada;
 - A validação cruzada (*cross-validation*) é uma técnica estatística usada para avaliar a generalização de um modelo, garantindo que seu desempenho não dependa exclusivamente de uma divisão específica dos dados (treino/teste).
- Para garantir a confiabilidade dos resultados, foi utilizada a métrica MAE (*Mean Absolute Error*) — Erro Absoluto Médio, a MSE (*Mean Squared Error*) - Erro Quadrático Médio e o Cálculo Matemático do Tempo de Execução, para avaliar o desempenho da rede neural.

Foram utilizadas as métricas MAE (*Mean Absolute Error*) e MSE (*Mean Squared Error*) para avaliar o desempenho do modelo de rede neural. O MAE foi utilizado por ser de fácil interpretação, indicando o erro médio absoluto em unidades de casos, o que permite uma análise direta do desempenho preditivo do modelo em termos práticos de saúde pública. Já o MSE foi utilizado para penalizar de forma mais intensa os erros maiores, permitindo identificar a sensibilidade do modelo em relação a variações bruscas nos dados, comuns em surtos de dengue. A utilização conjunta dessas métricas possibilitou uma avaliação equilibrada entre precisão média e impacto de outliers nos resultados do modelo.

3.3 Configuração da Rede Neural

3.3.1 Configuração da Rede Neural MLP

A configuração utilizada para a rede neural MLP foi a seguinte, determinada após a realização de diversos testes e experimentações sistemáticas para identificar os parâmetros que melhor equilibrassem desempenho e estabilidade durante o treinamento:

- **Camadas:** 256, 128 e 64 neurônios, respectivamente, distribuídos em três camadas densas sequenciais;
- **Dropout:** 0,2 em cada camada, com o objetivo de reduzir *overfitting* durante o treinamento;
Regularização L2: 1e-4 aplicada em cada camada, auxiliando no controle da complexidade do modelo;
- **Taxa de aprendizado (*learning rate*):** 0,0005, ajustada para proporcionar uma aprendizagem estável;
- **Épocas:** 150;
- **Batch size:** 32.

Para prevenir *overfitting* e otimizar o treinamento, foram utilizadas as seguintes estratégias:

- **Early Stopping:**
 - *monitor: val_loss;*
 - *patience: 15;*
 - *restore_best_weights: True;*
 - *verbose: 1;*
 - *min_delta: 0,001.*
- **ModelCheckpoint:**
 - caminho de salvamento: 'Dados/melhor_modelo_mensal.keras';
 - *monitor: val_loss;*
 - *save_best_only: True;*
 - *verbose: 1.*

3.3.2 Configuração da Rede Neural LSTM

A configuração utilizada para a rede neural LSTM foi definida a partir de experimentações e ajustes sucessivos, de forma a equilibrar capacidade de generalização, estabilidade e desempenho no treinamento para previsão de séries temporais de casos de dengue:

- **Camadas**
 - 1ª camada: LSTM bidirecional com 128 unidades, com retorno de sequências (*return_sequences=True*), aplicada regularização L2 (1e-4);
 - *Dropout* de 0,5 aplicado após a 1ª camada para redução de *overfitting*;
 - 2ª camada: LSTM bidirecional com 64 unidades, também com regularização L2 (1e-4);
 - *Dropout* de 0,4 aplicado após a 2ª camada;
 - Camada densa com 32 neurônios e ativação ReLU;
 - *Dropout* de 0,3 após a camada densa;
 - Camada de saída com 1 neurônio e ativação linear (*linear*), para previsão do valor contínuo de casos.
- **Taxa de aprendizado (*learning rate*): 0.0005**, ajustada para garantir estabilidade no gradiente durante o treinamento.
- **Épocas: 150**;
- **Batch size: 64** (ajustado para séries temporais).

Para prevenção de *overfitting* e otimização do processo de treinamento, foram aplicadas as seguintes estratégias:

- **Early Stopping:**
 - *monitor: val_loss;*
 - *patience: 15;*
 - *restore_best_weights: True;*
 - *verbose: 1;*
 - *min_delta: 0,001.*
- **ModelCheckpoint:**
 - *monitor: val_loss;*
 - *save_best_only: True;*
 - *verbose: 1.*

3.4 Ambiente Computacional

Para a execução dos experimentos, foi utilizado o seguinte ambiente computacional:

- **Processador:** AMD Ryzen 7 5700U;
- **Memória RAM:** 8GB DDR4;
- **Sistema Operacional:** Windows 11;
- **Placa de Vídeo (GPU):** Placa gráfica AMD Radeon™;
- **Armazenamento:** 256GB SSD;
- **Ambiente de Desenvolvimento:**
 - **Visual Studio Code** para desenvolvimento dos códigos, organização do projeto e execução dos scripts Python;
 - **MySQL Workbench** para criação e gerenciamento do banco de dados;
 - **GitHub** para versionamento e hospedagem do repositório público do projeto.

Devido à limitação da memória RAM durante o desenvolvimento do código, foi necessário utilizar exclusivamente a memória virtual (*swap*) alocada do SSD, resultando em um uso contínuo de 19 GB. Essa abordagem foi adotada para contornar a restrição de capacidade da RAM.

3.5 Aplicação da Rede Neural

Para a construção e aplicação da rede neural, foram seguidas as etapas descritas abaixo:

3.5.1 Carregamento e Pré-processamento dos Dados

- **Carregamento dos Dados:**
 - A primeira fase envolve a importação de dados de um banco de dados MySQL através do uso de SQLAlchemy e Pandas. A função `carregar_dados()` estabelece uma conexão com o banco de dados e executa uma consulta para obter os dados históricos de dengue. Caso a consulta não retorne dados ou ocorra um erro, o processo é interrompido, e uma mensagem de erro é registrada.
- **Pré-Processamento dos Dados:**
 - **Seleção das Colunas:** A função `processar_dados()` seleciona as colunas relevantes, incluindo dados sobre o estado, mês, ano, temperatura, precipitação, casos e mortes;
 - **Codificação de Variáveis Categóricas:** A função `pd.get_dummies()` converte as variáveis Estado e Mês em variáveis dummy, tornando-as em variáveis categóricas numéricas;
 - **Escalonamento de Dados:** As variáveis preditoras (X) são escaladas utilizando o *StandardScaler* para garantir que todas as variáveis possuam a mesma magnitude, enquanto os valores de resposta (y) são escalados com o *MinMaxScaler*.

3.5.2 Construção e Treinamento do Modelo Preditivo

- **Rede Neural MLP (*Multilayer Perceptron*)**

- A função `criar_e_treinar_modelo()` é responsável por construir e treinar o modelo de rede neural. A arquitetura é composta por três camadas ocultas densas que utilizam a função de ativação ReLU (*Rectified Linear Unit*). Para mitigar o *overfitting*, a arquitetura incorpora camadas de *dropout* e regularização L2 em suas camadas densas.

- **Rede Neural LSTM (*Long Short-Term Memory*)**

- A função `criar_modelo_lstm()` constrói um modelo com arquitetura especializada para análise de séries temporais. São utilizadas camadas LSTM bidirecionais, que processam a informação em ambas as direções (passado para o futuro e futuro para o passado). Essa abordagem aprimora a capacidade do modelo de capturar padrões temporais complexos e dependências de longo prazo nos dados.

- **Estratégias e Processo de Treinamento**

Para garantir a eficiência e a generalização dos modelos, foram adotadas as seguintes estratégias:

- *Early Stopping*: É empregado um mecanismo de parada antecipada que monitora a perda de validação (*validation loss*). O treinamento é interrompido automaticamente caso o desempenho no conjunto de validação não apresente melhora após um número pré-definido de épocas (parâmetro de "paciência"), evitando o sobreajuste.
- *ModelCheckpoint*: Durante o treinamento, uma cópia do modelo é salva sempre que a perda de validação atinge um novo mínimo. Isso garante que a versão com o melhor desempenho seja preservada, independentemente de onde o treinamento pare.

- **Divisão e Treinamento dos Dados**

- Ambos os modelos são treinados por um número definido de iterações (épocas), com os pesos sendo atualizados após o processamento de cada lote de dados (*batch*). O conjunto de dados é dividido em subconjuntos de treino e validação. É fundamental destacar que, no caso do modelo LSTM, a divisão dos dados é realizada com o parâmetro *shuffle=False*, preservando

a ordem cronológica que é essencial para a integridade das séries temporais.

- **Logs automáticos para rastreabilidade**

Para garantir a completa rastreabilidade e a transparência do pipeline de previsão, foi implementado um sistema de registro de eventos (*logs*) com o uso da biblioteca logging do *Python*. Este mecanismo captura informações em todas as etapas críticas do fluxo de trabalho, incluindo o carregamento dos dados a partir do banco MySQL, o pré-processamento, o treinamento dos modelos e a geração das previsões finais.

As mensagens de log, geradas em tempo real, documentam eventos-chave, tais como:

- A confirmação do sucesso no carregamento dos dados;
- Alertas para anomalias, como a ausência de colunas esperadas;
- Registros de erros de conexão ou de processamento;
- A confirmação do salvamento de métricas e previsões em arquivos CSV.

A implementação deste sistema de monitoramento automatizado demonstrou-se fundamental para agilizar o diagnóstico de falhas e assegurar a integridade e a reprodutibilidade de cada execução dos modelos.

3.5.3 Previsão e Avaliação

- **Previsão:**

- **MLP:** Depois do treinamento, o modelo foi desenvolvido para fazer previsões mensais de casos de dengue. A função `fazer_previsoes_mensais()` percorre todos os estados e meses, cria as entradas necessárias para cada combinação, normaliza os dados e realiza as previsões de casos.
- **LSTM:** A função `fazer_previsoes_lstm()` realiza previsões considerando o contexto temporal, utilizando sequências históricas de 3 períodos para cada previsão. O processo mantém o histórico de cada estado separadamente, garantindo que as dependências temporais sejam preservadas durante a previsão.

- **Avaliação:**

- **Erro Quadrático Médio (MSE):**

- Para avaliar a qualidade das previsões realizadas pelos modelos, foi utilizado o MSE que mede a média dos quadrados das diferenças entre os valores previstos e os valores reais. Esta métrica penaliza erros grandes de forma mais severa devido ao termo quadrático, sendo fundamental para identificar a sensibilidade do modelo em relação a variações bruscas nos dados, comuns em surtos de dengue;
- **Erro Absoluto Médio (MAE):**
 - O MAE foi empregado como métrica complementar por ser menos sensível a *outliers* do que o MSE e fornecer uma interpretação mais direta dos erros em unidades da variável prevista. Esta métrica mede a média dos valores absolutos das diferenças entre os valores reais e previstos, permitindo uma análise direta do desempenho preditivo do modelo em termos práticos de saúde pública;
- **Medição do Tempo de Execução:**
 - Para monitorar o desempenho de ambos os modelos preditivos, foi calculado o tempo total de execução do *pipeline* de processamento, treinamento e previsão utilizando a função `time.time()` da biblioteca padrão *Python*. Os tempos são salvos em arquivos CSV específicos para cada modelo: `tempo_execucao.csv` (MLP) e `tempo_execucao_lstm.csv` (LSTM).

3.5.4 Armazenamento das previsões mensais em CSV

Após o treinamento do modelo, foi realizada a etapa de previsão de casos e mortes de dengue para todos os estados brasileiros, mês a mês, para o ano de 2024. Esse processo foi automatizado por meio da função `fazer_previsoes_mensais()`, que percorre sistematicamente cada combinação de estado e mês existente no banco de dados, gerando entradas de previsão para o modelo.

Cada entrada é normalizada utilizando o *scaler* previamente ajustado, passada para o modelo treinado, e o resultado da previsão (número de casos) é desnormalizado utilizando o mesmo *scaler* para retornar os valores para a escala original.

As previsões geradas são armazenadas em uma lista de dicionários, onde cada dicionário contém o ano, mês, estado, o número de casos previstos e o número de mortes previstas para aquela combinação.

Após a geração das previsões para todas as combinações, a lista é convertida em um *DataFrame* do Pandas, permitindo a manipulação estruturada dos dados. Em seguida, o *DataFrame* é exportado para um arquivo CSV nomeado `previsoes_mensais.csv`, garantindo o armazenamento de todas as previsões de forma organizada e acessível para

análises posteriores. Caso o arquivo já exista, o sistema realiza a leitura do arquivo existente, concatena os novos registros e salva novamente, mantendo um histórico cumulativo das previsões.

Além disso, para controle de auditoria e rastreabilidade, é adicionada uma coluna timestamp no CSV, registrando o momento exato em que as previsões foram realizadas.

4 Resultados

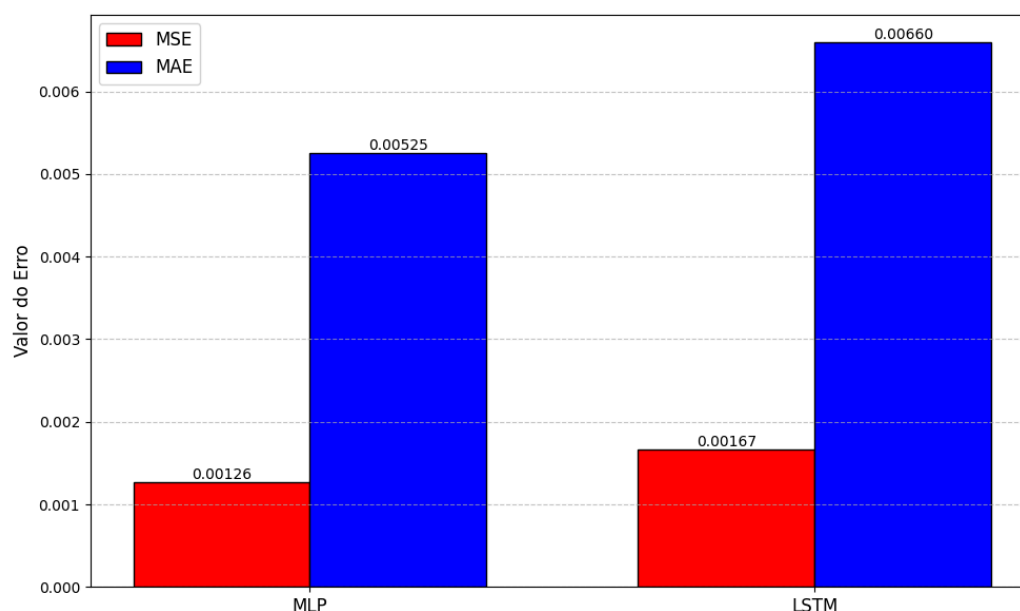
Os modelos de redes neurais MLP (*Multilayer Perceptron*) e LSTM (*Long Short-Term Memory*) foram treinados e avaliados utilizando dados epidemiológicos e climáticos do período de 2014 a 2023. Os resultados obtidos demonstram a viabilidade da aplicação de técnicas de aprendizado de máquina na predição de casos de dengue no Brasil.

4.1 Avaliação do Desempenho dos Modelos

4.1.1 Comparação das Métricas de Erro

A análise comparativa entre os modelos MLP e LSTM revelou diferenças significativas no desempenho preditivo, conforme ilustrado no Gráfico 1. As métricas utilizadas foram o Erro Quadrático Médio (MSE) e o Erro Absoluto Médio (MAE), onde valores mais próximos de zero indicam melhor performance do modelo.

Gráfico 1 - Comparação de Erros entre MLP e LSTM



Fonte - Autor, 2025

Com os experimentos realizados foi possível comparar de forma sistemática os modelos MLP e LSTM aplicados à previsão dos dados analisados. Observou-se que o modelo MLP apresentou uma superioridade consistente, alcançando um MSE 24,5% menor (0,00126 em comparação a 0,00167) e um MAE 20,5% menor (0,00525 contra 0,00660).

É importante destacar que tanto o MSE (*Mean Squared Error*) quanto o MAE (*Mean Absolute Error*) são métricas de erro, nas quais valores mais próximos de zero indicam maior precisão preditiva. O MSE penaliza mais severamente erros grandes devido ao termo quadrático, enquanto o MAE fornece uma medida mais robusta para tratar todos os erros de forma linear. A superioridade do MLP em ambas as métricas indica não apenas maior acurácia global, mas também melhor comportamento em relação a outliers nos dados epidemiológicos.

Com os experimentos realizados foi possível comparar de forma sistemática os modelos MLP e LSTM aplicados à previsão dos dados analisados. Observou-se que o modelo MLP apresentou uma superioridade consistente, alcançando um MSE 24,5% menor (0,00126 em comparação a 0,00167) e um MAE 20,5% menor (0,00525 contra 0,00660).

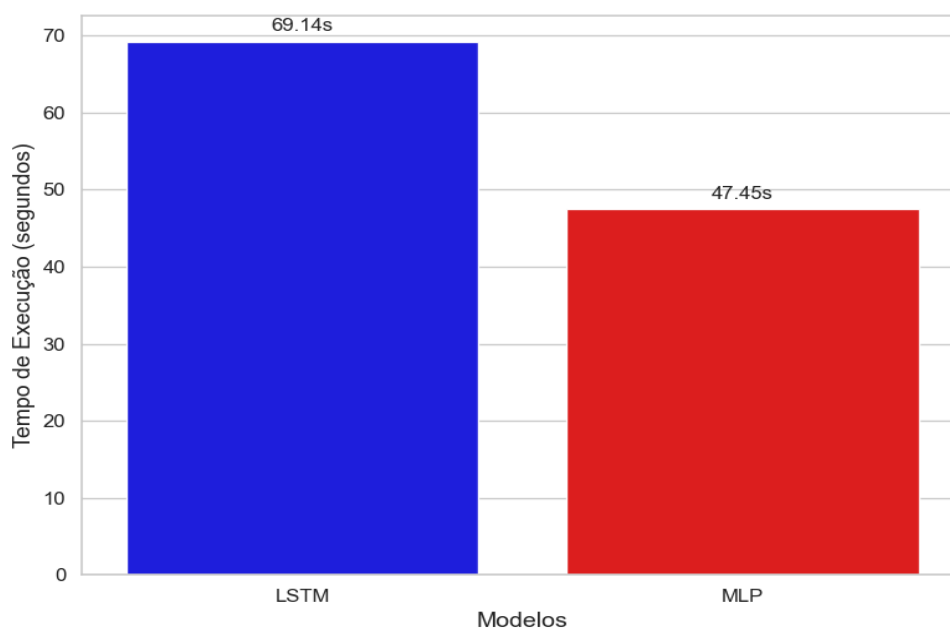
4.1.2 Análise da Superioridade do Modelo MLP

A superioridade do modelo MLP em relação ao LSTM pode ser explicada pelas características específicas dos padrões epidemiológicos da dengue no contexto brasileiro. Embora redes LSTM sejam tradicionalmente aplicadas a problemas de séries temporais devido à sua capacidade de capturar dependências de longo prazo, os resultados deste estudo indicam que o fenômeno analisado apresenta particularidades que favoreceram o desempenho do MLP. Em primeiro lugar, a série histórica utilizada é relativamente curta (2014–2023), o que limita o potencial da LSTM de explorar relações temporais extensas. Além disso, os dados de casos de dengue apresentam alta variabilidade sazonal e forte presença de picos abruptos decorrentes de surtos, o que pode ter levado a rede LSTM a confundir ruídos e outliers como padrões relevantes, reduzindo sua precisão.

4.1.3 Eficiência Computacional

A análise do tempo de execução, apresentada no Gráfico 2, revela diferenças substanciais na eficiência computacional entre os modelos.

Gráfico 2 – Comparação do tempo de Execução dos Modelos



Fonte - Autor, 2025

Os resultados indicam uma vantagem clara para o modelo MLP também neste quesito. O MLP concluiu seu processo de execução em 47,45 segundos, enquanto o modelo LSTM necessitou de 69,14 segundos. Isso representa uma economia de tempo de aproximadamente 31,4% em favor do MLP.

Essa diferença de desempenho é atribuída à complexidade arquitetônica de cada modelo. O MLP possui uma estrutura feedforward, onde os dados fluem em uma única direção, resultando em um processo computacional mais direto e rápido. Em contrapartida, o LSTM é uma rede recorrente projetada para lidar com sequências, possuindo células de memória e portões internos que mantêm o contexto temporal. Embora poderosa para dados sequenciais, essa estrutura exige um número maior de cálculos a cada passo, tornando-a inerentemente mais lenta e computacionalmente mais custosa que a arquitetura mais simples do MLP.

5 Considerações Finais

Este trabalho desenvolveu e validou modelos de redes neurais artificiais para predição de casos de dengue no Brasil, utilizando arquiteturas MLP (*Multilayer Perceptron*) e LSTM (*Long Short-Term Memory*) com dados epidemiológicos e climáticos do período de 2014 a 2023. A pesquisa consolidou uma metodologia robusta para processamento de dados heterogêneos de múltiplas fontes (DATASUS e INMET), criando scripts automatizados para extração, limpeza e transformação dos dados. Embora o modelo LSTM seja teoricamente robusto para dependências de longo prazo, o modelo MLP demonstrou ser mais eficaz e eficiente para esta série de dados específica, possivelmente devido à sua menor complexidade e maior capacidade de lidar com a variabilidade e os picos abruptos dos dados epidemiológicos.

Durante o desenvolvimento, enfrentou-se desafios significativos relacionados à heterogeneidade dos dados, que apresentaram diferentes formatos de data, granularidades temporais distintas e múltiplas estações meteorológicas. As limitações computacionais do ambiente de desenvolvimento foram contornadas através de adaptações no código e uso de memória virtual, evidenciando a importância do planejamento de recursos em projetos de aprendizado de máquina. A aplicação de técnicas de validação cruzada, *early stopping* e *model checkpoint* garantiu a robustez dos modelos, enquanto as métricas MAE e MSE forneceram avaliações complementares do desempenho, permitindo análises da precisão média e sensibilidade a outliers.

Os modelos desenvolvidos geram previsões mensais para todos os estados brasileiros, criando um arcabouço técnico que pode ser integrado a sistemas de vigilância epidemiológica. A capacidade de antecipar cenários permite aos gestores de saúde pública o planejamento estratégico de recursos, campanhas preventivas direcionadas e preparação da rede assistencial. A consideração de variáveis climáticas no modelo reforça a importância dos fatores ambientais na dinâmica de transmissão da dengue, oferecendo subsídios científicos para políticas públicas que considerem as mudanças climáticas como determinante epidemiológico.

Para trabalhos futuros, recomenda-se investir na calibração e otimização dos parâmetros, incluir novas variáveis como densidade populacional, indicadores socioeconômicos e dados de saneamento básico, além de experimentar arquiteturas híbridas e técnicas de ensemble learning. A migração para ambientes de computação em nuvem permitirá o desenvolvimento de modelos mais complexos, enquanto a criação de interfaces interativas e *dashboards* facilitará a aplicação prática pelos gestores de saúde. Este trabalho estabelece uma base sólida para o uso de inteligência artificial na predição de dengue, demonstrando o potencial transformador das técnicas de aprendizado de máquina na epidemiologia e sua contribuição para o fortalecimento dos sistemas de vigilância epidemiológica no Brasil.

6 - Referências Bibliográficas

ALAKE, Richmond. **Explicação das funções de perda no aprendizado de máquina.** 2024. Disponível em: <https://www.datacamp.com/pt/tutorial/loss-function-in-machine-learning>. Acesso em: 19 maio 2025.

ALI, Moez. **Introdução às funções de ativação em redes neurais.** 2024. Disponível em: <https://www.datacamp.com/pt/tutorial/introduction-to-activation-functions-in-neural-networks>. Acesso em: 19 maio 2025.

BELTA.IO. **Supervised Machine Learning.** Disponível em: <https://blog.balta.io/azure-machine-learning-tipos/#:~:text=Uma%20feature%20é%20uma%20variável,e%20fazer%20previsões%20ou%20decisões..> Acesso em: 16 jul. 2025.

CASTRO, Angélica F. *et al.* **Utilização da linguagem Python na análise de dados de dengue em Recife, capital do estado de Pernambuco, Nordeste do Brasil e definição**

de linha de tendência usando *Machine Learning*. 2022. Disponível em: https://www.researchgate.net/publication/361989988_Utilizacao_da_linguagem_Python_na_analise_de_dados_de_dengue_em_Recife_capital_do_estado_de_Pernambuco_Nordeste_do_Brasil_e_definicao_de_linha_de_tendencia_usando_Machine_Learning_Use_of_Python_langua. Acesso em: 10 mar. 2025.

CEARÁ. **Secretaria da Saúde. Calor e período chuvoso aumentam riscos de proliferação de mosquito transmissor da dengue; conheça os sintomas e saiba como se proteger.** 2024. Disponível em: <https://www.saude.ce.gov.br/2024/02/26/calor-periodo-chuvoso-riscos-proliferacao-mosquito-transmissor-dengue-sintomas/>. Acesso em: 20 mar. 2025.

CLÉBIO DE OLIVEIRA JÚNIOR. **Comunidade *Data Hackers Follow publication Top highlight* Métricas para Regressão: Entendendo as métricas R^2 , MAE, MAPE, MSE e RMSE.** 2021. Disponível em: <https://medium.com/data-hackers/prevendo-n%C3%BAmoros-entendendo-m%C3%A9tricas-de-regress%C3%A3o-35545e011e70>. Acesso em: 03 jul. 2025.

EVSUKOFF, A. G. **Inteligência computacional: Fundamentos e aplicações.** 1. ed. ed. Rio de Janeiro: E-papers, 2020.

FERNANDES, Luiz Gustavo L.; PORTUGAL, Marcelo Savino; NAVAUX, Philippe O. A. **Previsão de Séries de Tempo: Redes Neurais Artificiais e Modelos Estruturais.** 1995. Disponível em: <https://gmap.pucrs.br/files/publications/articles/5afca684be812ed8e2a901173d20e6b6.pdf>. Acesso em: 20 fev. 2025.

GRÜBLER, Murillo (org.). ***Stories Stats aibrazil Home About Entendendo o funcionamento de uma Rede Neural Artificial.*** 2018. Disponível em: <https://medium.com/brasil-ai/entendendo-o-funcionamento-de-uma-rede-neural-artificial-4463fcf44dd0>. Acesso em: 18 mar. 2025.

GERON, Aurélien. ***Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.*** 2. ed. Sebastopol: O'Reilly Media, 2019

H20.AI. **O que é uma variável alvo em aprendizado de máquina?** Disponível em: https://h2o-ai.translate.google/wiki/target-variable/?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=rq#:~:text=A%20variável-alvo%20é%20a,restante%20do%20conjunto%20de%20dados.. Acesso em: 16 jul. 2025.

HUDSON BARROSO LUCAS (org.). **Regularização em *Machine Learning*: Técnicas e Benefícios para Modelos Mais Robustos.** 2024. Disponível em: <https://iacomcafe.com.br/regularizacao-machine-learning-l1-l2-dropout-early-stopping-batch-normalization/>. Acesso em: 04 jul. 2025.

IBM. **O que é uma rede neural?: redes neurais versus deep learning.** Redes neurais versus deep learning. Disponível em: <https://www.ibm.com/br-pt/think/topics/neural-networks>. Acesso em: 25 mar. 2025.

LEITE, Eduardo Soares. **Análise temporal da relação entre dengue e variáveis climáticas na cidade de Uberlândia – MG**. 2023. 101 f. Monografia (Especialização) - Curso de Pós-Graduação em Geografia: Análise, Planejamento e Gestão Ambiental, Universidade Federal de Uberlândia, Uberlândia, 2023. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/37954/3/An%C3%A1liseTemporalRela%C3%A7%C3%A3o.pdf>. Acesso em: 20 mar. 2025.

LEITE, Tiago M. **Redes Neurais, Perceptron Multicamadas e o Algoritmo Backpropagation**. 2018. Disponível em: <https://medium.com/ensina-ai/redes-neurais-perceptron-multicamadas-e-o-algoritmo-back-propagation-eaf89778f5b8>. Acesso em: 18 mar. 2025.

MAJEED, Mokhalad A.. **A Deep Learning Approach for Dengue Fever Prediction in Malaysia Using LSTM with Spatial Attention**. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10002017/pdf/ijerph-20-04130.pdf>. Acesso em: 21 jul. 2025.

MATTIOLI, Fernando Eduardo Rezende; ANDRADE, Rogério Bernardes; ESTEVEZ, Emanuel Tobias. Predição de casos de dengue utilizando redes neurais artificiais. *Jetma*, Uberaba, p. 8-12, jan. 2017.

MGHAZLI, Yá-Sin Barcelos. **REDES NEURAIS PARA PREDIÇÃO DE SERIES TEMPORAIS**. 2021. Disponível em: <https://pantheon.ufrj.br/bitstream/11422/19468/1/YBMghazli.pdf>. Acesso em: 16 jul. 2025.

MINISTÉRIO DA AGRICULTURA E PECUÁRIA. **DADOS HISTÓRICOS ANUAIS**. Disponível em: <https://portal.inmet.gov.br/dadoshistoricos>. Acesso em: 04 fev. 2025.

MINISTÉRIO DA SAÚDE/SVSA. **Dengue - Notificações registradas no Sistema de Informação de Agravos de Notificação - Brasil**. Disponível em: https://tabnet.saude.sp.gov.br/deftohtm.exe?tabnet/sim_ccd.def. Acesso em: 25 mar. 2025.

MINISTÉRIO DA SAÚDE/SVSA. **Dengue - Notificações registradas no Sistema de Informação de Agravos de Notificação - Brasil**. 2024. Disponível em: <http://tabnet.datasus.gov.br/cgi/deftohtm.exe?sinannet/cnv/denguebbr.def>. Acesso em: 25 mar. 2025.

PHILIPS. **Análise preditiva em saúde: três exemplos do mundo real**. 2020. Disponível em: <https://www.philips.com.br/a-w/about/news/archive/standard/article/2020/20200604-predictive-analytics-in-healthcare-three-real-world-examples.html>. Acesso em: 19 mar. 2025.

PISHRO-NIK, Hossein. **Introduction to Probability, Statistics, and Random Processes**. *Sunderland*: Kappa Research, Llc, 2014.

REIS, Letícia Fernandes dos. **IDENTIFICAÇÃO DE POTENCIAIS FOCOS DE DENGUE USANDO REDES NEURAIS ARTIFICIAIS COM DADOS DE SENSORIAMENTO REMOTO**. 2023. 44 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação,

Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Betim, Betim, 2023.

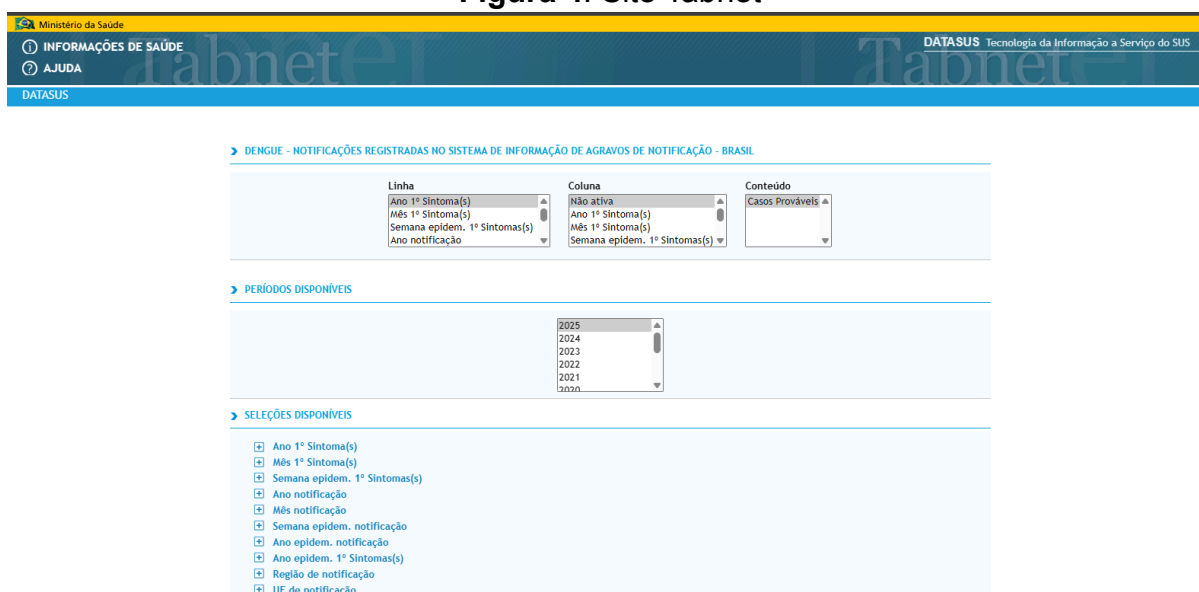
SIQUEIRA, Ionara Santos *et al.* **A relação da incidência de casos de dengue com a precipitação na área urbana de Belém-PA, 2007 a 2011**, através de modelos multivariados de séries temporais. 2018. Disponível em: <https://www.scielo.br/j/rbmet/a/nzvxBYjzNXtK7CCjqjXFP6q/#>. Acesso em: 19 mar. 2025.

SILVA, Caíque Quaresma. **Redes neurais aplicadas no reconhecimento de símbolos matemáticos manuscritos online**. 2019. 54 f. Trabalho de Conclusão de Curso (Graduação em Matemática Aplicada) – Universidade de São Paulo, São Paulo, 2019. Disponível em: <https://www.ime.usp.br/~mapweb/tcc/2019/CaiqueQuaresmaV2.pdf>. Acesso em: 20 mar. 2025.

APÊNDICE A - Tutorial Dados Saúde

Ao acessar o Tabnet encontramos três menus principais de seleção, como podemos ver na Figura 4, que funcionam da seguinte forma: no menu “DENGUE - Notificações registradas no Sistema de Informação de Agravos de Notificação - Brasil”, temos a possibilidade de definir o que constará na linha e na coluna, como por exemplo: "Ano de Notificação" na linha e "UF de residência" na coluna. No menu seguinte, “Períodos Disponíveis”, é feita a seleção do ano desejado, não sendo sempre necessária a sua utilização, a depender da necessidade. Após isso, há o último menu, “Seleções Disponíveis”, onde ficam todas as variáveis com as quais podemos trabalhar nesse painel,

Figura 4: Site Tabnet



Fonte - Autor, 2025

Ao final da página do painel, encontram-se as opções de formatação, como mostrado na Figura 5. Nelas, define-se como os dados serão apresentados. A seleção feita nesta etapa não terá influência na forma como os dados serão baixados, apenas em como serão exibidos na tela seguinte. A única opção de real importância é “Exibir linhas zeradas”, que deve ser selecionada. Após isso, pode-se clicar no botão “Mostra”.

Figura 5: Site Tabnet

The screenshot shows the Tabnet interface with a list of exams on the left and formatting options on the right. The exams listed are: Exame de RT-PCR, Sorotipo, Exame de Histopatologia, Exame de Imunohistoquímica, and Ocorreu hospitalização. The formatting options include checkboxes for 'Ordenar pelos valores da coluna' and 'Exibir linhas zeradas', and radio buttons for 'Formato' with options 'Tabela com bordas' (selected), 'Texto pré-formatado', and 'Colunas separadas por ";"'. There are 'Mostra' and 'Limpa' buttons at the bottom. The footer text reads: 'Fonte: Ministério da Saúde/SVSA - Sistema de Informação de Agravos de Notificação - Sinan Net'.

Fonte - Autor, 2025

Neste momento, é exibida a tela com os dados, conforme ilustra a Figura 6. Nela, realiza-se uma pré-análise do conteúdo, a fim de verificar se os dados são os necessários e se a configuração utilizada está correta. Tendo certeza de que as informações estão adequadas, clica-se no botão “Copiar como .CSV” e, então, o download do arquivo será iniciado.

Figura 6: Tabela de dados Tabnet

▶ **DEQUE - NOTIFICAÇÕES REGISTRADAS NO SISTEMA DE AGRAVOS DE NOTIFICAÇÃO - BRASIL**

Casos Prováveis por UF de residência segundo mês notificação
Período: 2016

Mês notificação	RO	AC	AM	RR	PA	AP	TO	MA	PI	CE	RN	PB	PE	AL	SE	BA	MG	ES	RJ	SP	PR	SC	RS	MS	MT
TOTAL	7.918	2.195	7.586	212	10.945	1.815	7.895	23.928	5.247	49.721	57.012	35.796	62.553	19.454	3.518	67.748	528.985	43.146	88.316	202.587	62.625	5.088	3.180	46.890	21.212
Jan/16	1.716	316	444	13	1.271	88	1.546	2.046	280	1.188	3.274	4.279	16.324	3.099	437	6.891	50.384	10.064	16.369	31.384	8.234	277	133	15.927	6.212
Fevereiro	2.277	767	925	14	2.202	263	2.149	4.484	469	2.759	15.546	6.245	17.745	3.504	1.101	20.503	128.388	10.881	16.497	52.426	16.332	1.240	340	14.881	6.212
Março	1.720	367	1.343	48	2.124	410	1.444	5.786	574	4.560	19.804	9.349	13.078	3.598	414	20.869	161.457	9.800	17.427	56.611	19.191	1.341	1.130	9.499	13.212
Abril	847	235	1.710	10	1.723	311	913	5.989	1.562	8.709	6.994	6.652	5.477	2.862	433	8.935	125.279	6.475	18.464	38.700	14.285	1.185	1.067	4.725	11.212
Mai	290	103	1.005	9	1.117	201	830	2.525	1.039	11.772	4.850	4.321	3.237	2.158	168	3.884	45.580	2.234	10.260	15.151	2.774	275	351	906	11.212
Junho	112	67	777	9	521	86	324	1.014	542	7.740	2.932	2.075	1.953	1.341	92	1.710	8.568	600	3.662	3.018	406	25	28	235	11.212
Julho	112	64	388	15	484	72	131	788	243	4.810	1.565	1.256	1.330	936	126	930	1.763	413	1.934	1.102	214	21	14	95	11.212
Agosto	29	46	228	36	346	100	93	598	184	2.363	816	350	1.159	655	195	725	1.034	360	885	609	135	19	16	110	11.212
Setembro	40	40	186	25	263	92	56	201	97	1.388	412	195	777	385	87	582	796	341	565	547	141	16	10	111	11.212
Outubro	43	41	177	14	290	91	74	125	85	1.055	217	310	597	222	100	745	868	325	487	785	237	22	19	113	11.212
Novembro	154	61	224	11	269	57	147	213	67	1.680	227	126	538	146	73	824	1.125	508	721	1.124	356	29	23	103	11.212
Dezembro	578	98	199	8	415	44	188	229	85	1.097	275	68	438	168	92	1.169	1.763	975	1.045	1.130	320	38	49	185	11.212

Fonte: Ministério da Saúde/SISA - Sistema de Informação de Agravos de Notificação - Sinan Net

Notas:

1. Para os casos prováveis foram incluídas todas notificações, exceto casos descartados.
2. Os dados de dados de dengue atualizados compõem um banco único a partir de 2014, podendo haver pequenas divergências com os dados disponibilizados pelo CGA88 (Coordenação Geral de Vigilância de Arboviroses - CGA88) em sua série histórica, que para tal, realiza análise ano a ano, com a base completa para cada ano atualizado.
3. Para tabular os casos graves (classificação final igual a dengue com complicações, febre hemorrágica de dengue, síndrome do choque de dengue, dengue com sinais de alarme e dengue grave) é necessário considerar também o critério de confirmação (laboratorial e clínico-epidemiológico).
4. A partir de 2020 o estado do Espírito Santo passou a utilizar o sistema e-SIS Vigilância em Saúde. Portanto, para os casos de Arboviroses oriundos do Espírito Santo foram considerados apenas os dados disponibilizados pelo Sinan online (dengue e chikungunya) e Sinan Net (Zika).
5. Períodos Disponíveis ou período - Correspondem aos anos de notificação dos casos e semana epidemiológica, em cada período pode apresentar notificações com data de notificação do ano anterior (semana epidemiológica 52 ou 53) e posterior (semana epidemiológica 01).
6. Para cálculo da incidência recomenda-se utilizar fontes de veracidade.
7. Dados de 2014 atualizados em 11/07/2015.
8. Dados de 2015 atualizados em 27/09/2015.
9. Dados de 2016 atualizados em 18/07/2016.
10. Dados de 2017 atualizados em 18/07/2016.
11. Dados de 2018 atualizados em 01/10/2019.
12. Dados de 2019 atualizados em 10/07/2020.
13. Dados de 2020 atualizados em 10/07/2021.
14. Dados de 2021 atualizados em 12/07/2022.
15. Dados de 2022 atualizados em 16/07/2023.
16. Dados de 2023 atualizados em 04/06/2024 às 07 horas.
17. Dados de 2024 atualizados em 21/05/2025 às 10:00 horas, sujeitos à revisão.
18. Dados de 2025 atualizados em 21/07/2025 às 10:30 horas, sujeitos à revisão.

* Dados disponibilizados no TABNET julho de 2025.

Legenda:

- Dado numérico igual a 0 não resultante de arredondamento.
- 0,0.0 - Dado numérico igual a 0 resultante de arredondamento de um dado originalmente positivo.

CÓPIA COMO .CSV

CÓPIA PARA TABNET

Fonte - Autor, 2025

APÊNDICE B - Tutorial de Dados Climáticos

Inicialmente, se acessa o portal oficial do Instituto Nacional de Meteorologia, conforme ilustra a Figura 7. Na plataforma, encontram-se dados para todo o Brasil a partir do ano 2000. Ao clicar no ano desejado, inicia-se o download de arquivos compactados (em formato ZIP), que contém arquivos CSV com as informações climáticas horárias de todas as estações meteorológicas do país para o período selecionado.

Figura 7: Site INMET

COMUNICAÇÃO
ACESSOS À INFORMAÇÃO
PARTICIPAÇÃO
LEGISLAÇÃO
ÓRGÃOS DO GOVERNO

Ir para o conteúdo | Ir para o menu | Ir para o rodapé

SOBRE O INMET | IMPRENSA | MAPA DO SITE | FALE CONOSCO | VÍDEO

Instituto Nacional de Meteorologia

MINISTÉRIO DA AGRICULTURA E PECUÁRIA

Tempo
Clima
Dados Meteorológicos
Satélites
Risco de Incêndio
Previsão Numérica
Sisdiagro
CIVEM
Publicações
Sobre
Informações

DADOS HISTÓRICOS ANUAIS

ANO 2000 (AUTOMÁTICA)

ANO 2001 (AUTOMÁTICA)

ANO 2002 (AUTOMÁTICA)

ANO 2003 (AUTOMÁTICA)

ANO 2004 (AUTOMÁTICA)

ANO 2005 (AUTOMÁTICA)

ANO 2006 (AUTOMÁTICA)

ANO 2007 (AUTOMÁTICA)

Fonte - Autor, 2025