

Comparação entre algoritmos de Machine Learning na predição de casos de Dengue

Mikael Johnatan da Silva Azevedo¹
mikael.azevedo@alunos.ufersa.edu.br

João Pedro Pereira Frutuoso¹
joao.frutuoso@ufersa.edu.br

Amanda Gondin de Oliveira¹
amandagondim@ufersa.edu.br

Angélica Félix de Castro¹
angelica@ufersa.edu.br

Paulo Gabriel Gadelha Queiroz¹
pgabriel@ufersa.edu.br

Universidade Federal Rural do Semi-Árido¹
R. Francisco Mota, 575 - Presidente Costa e Silva, Mossoró - RN, 59625-900

RESUMO

A dengue é uma arbovirose que afeta significativamente a saúde pública, especialmente em regiões tropicais e subtropicais como o Brasil, onde são frequentes os surtos epidêmicos. Antecipar sua disseminação é fundamental para ações preventivas e respostas ágeis dos serviços de saúde. Este estudo avaliou e comparou o desempenho de algoritmos de aprendizado de máquina (Machine Learning) e aprendizado profundo (Deep Learning) na previsão de casos de dengue, utilizando variáveis climáticas e epidemiológicas organizadas de forma temporal. Foram analisados modelos clássicos, como Random Forest, Support Vector Regression (SVR) e Multilayer Perceptron, bem como redes neurais profundas, como Long Short-Term Memory (LSTM) e Convolutional Neural Network (CNN). Os resultados indicaram que o modelo LSTM apresentou o melhor equilíbrio entre acurácia preditiva e tempo de execução, enquanto o SVR destacou-se como alternativa eficiente em cenários com limitações de recursos computacionais.

PALAVRAS-CHAVE. **Aprendizado de máquina, Dengue, Redes neurais.**

EST&AM – PO Analytics em Estatística e Aprendizado de Máquina; SA – PO na Área de Saúde; AS&DS – Análise e Ciência de Dados;

ABSTRACT

Dengue is an arboviral disease that significantly impacts public health, especially in tropical and subtropical regions such as Brazil, where epidemic outbreaks are frequent. Anticipating its spread is crucial for preventive measures and rapid health service responses. This study evaluated and compared the performance of machine learning (ML) and deep learning (DL) algorithms in forecasting dengue cases, using climatic and epidemiological variables organized in a temporal manner. Classical models such as Random Forest, Support Vector Regression (SVR), and Multilayer Perceptron were analyzed, along with deep neural networks such as Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN). The results indicated that the LSTM model offered the best balance between predictive accuracy and execution time, while SVR emerged as an efficient alternative in scenarios with limited computational resources.

KEYWORDS. **Machine learning, Dengue, Neural networks.**

EST&AM – Statistics and Probabilistic Models; SA - OR in Health; AS&DS – Data Science and Analytics;

1. Introdução

Dengue é uma das arboviroses predominantes nas regiões tropicais e subtropicais e, globalmente, representa um grande desafio para a saúde pública. A doença, transmitida pelo mosquito *Aedes aegypti*, pode evoluir para formas graves, como a dengue hemorrágica, resultando em hospitalização e podendo levar à morte. De acordo com a Organização Mundial da Saúde [OMS 2022], “a dengue é hoje a doença viral transmitida por vetores que mais cresce no mundo, com um aumento de 30 vezes na incidência nas últimas cinco décadas”. Estima-se que cerca de 390 milhões de infecções por dengue ocorram anualmente, das quais aproximadamente 96 milhões manifestam sintomas clínicos [Bhatt et al., 2013]. No Brasil, a doença é considerada endêmica, com surtos recorrentes que sobrecarregam os sistemas de saúde. Conforme aponta o Ministério da Saúde [BRASIL 2023], “as epidemias de dengue representam um dos maiores desafios para a gestão dos serviços públicos de saúde, exigindo ações integradas de vigilância, prevenção e controle”.

Diante desse cenário, a previsão de surtos de dengue é essencial para a adoção de medidas preventivas, como campanhas de conscientização, controle vetorial e alocação de recursos médicos. A vigilância epidemiológica tradicional baseia-se em dados históricos e no monitoramento em tempo real; no entanto, com o avanço da inteligência artificial, técnicas de Machine Learning (ML) e Deep Learning (DL) têm apresentado resultados promissores na modelagem preditiva da incidência de dengue, alcançando modelos capazes de prever surtos com meses de antecedência [Aleixo et al., 2022]. Esses algoritmos podem identificar padrões complexos fundamentais, auxiliando na antecipação de epidemia.

Neste estudo, foi realizada a avaliação do desempenho de diferentes modelos de aprendizado de máquina na predição de casos de dengue. Foram utilizados algoritmos clássicos de Machine Learning (ML), como Random Forest, Support Vector Machine Regressor e Multilayer Perceptron, além de arquiteturas de Deep Learning (DL), como Long Short-Term Memory e Convolutional Neural Network. O objetivo foi comparar a eficácia desses modelos, considerando diferentes parâmetros, a fim de identificar a abordagem com maior precisão na previsão da incidência da doença. Para tal, os modelos foram implementados com o uso das bibliotecas Scikit-Learn (para os modelos tradicionais) e TensorFlow (para as redes neurais profundas), sendo avaliados por meio de técnicas de validação cruzada e métricas robustas, como o Coeficiente de Determinação (R^2), o Erro Médio Quadrático (RMSE) e o teste estatístico de Friedman.

Dessa forma, os resultados deste trabalho podem contribuir para o desenvolvimento de sistemas de alerta precoce mais eficientes, auxiliando gestores públicos na tomada de decisões estratégicas para o combate à dengue. Além disso, a análise comparativa entre os algoritmos fornece dados concretos sobre as vantagens e limitações de cada método em problemas de predição epidemiológica.¹

Este artigo está estruturado da seguinte forma: na seção 2, "Fundamentação Teórica", apresentamos a contextualização do problema e uma revisão dos principais algoritmos de predição, bem como as métricas de avaliação utilizadas. A seção 3 descreve a metodologia adotada, abordando a coleta de dados, o processo de análise e pré-processamento dos dados e os parâmetros de execução dos algoritmos. Em seguida, na seção 4, são apresentados os resultados obtidos a partir da execução dos modelos. Por fim, a Seção 5 traz as conclusões do estudo e sugestões para trabalhos futuros.

¹ <https://scikit-learn.org/stable/documentation.html>
https://www.tensorflow.org/api_docs

2. Fundamentação Teórica

2.1. Contextualização do Problema

A dinâmica de transmissão da dengue está intrinsecamente associada a fatores climáticos, que influenciam diretamente o ciclo de vida do *Aedes aegypti* e, conseqüentemente, o potencial epidêmico da doença. Variáveis como temperatura média, precipitação acumulada e umidade relativa do ar afetam significativamente a reprodução do vetor — temperaturas entre 25°C e 30°C aceleram seu desenvolvimento larval, enquanto a precipitação cria criadouros ideais para a produção do mosquito [Beserra et al., 2009]. Como destacado por Morin et al. (2013), “mudanças nos padrões climáticos podem alterar significativamente a incidência de doenças transmitidas por vetores, inclusive antecipando picos epidêmicos”. Neste contexto, a captura precisa de dados meteorológicos é imprescindível para a construção de modelos preditivos confiáveis, pois possibilita quantificar essas relações de causalidade e identificar padrões temporais de risco potencial. Conforme destaca o relatório da Organização Mundial da Saúde [OMS 2017], “a vigilância integrada de fatores ambientais e epidemiológicos é crucial para o gerenciamento eficaz de surtos de dengue”. Assim, a integração de variáveis climáticas com dados epidemiológicos neste estudo abrange práticas avançadas de modelagem para doenças transmitidas por vetores e propõe uma abordagem científica validada, capaz de apoiar as autoridades sanitárias na antecipação de ações em cenários de crise, superando as limitações dos sistemas tradicionais de vigilância, que dependem exclusivamente de notificações em tempo real.

2.2. Algoritmos de Predição

Foram utilizados cinco algoritmos de aprendizado de máquina amplamente reconhecidos por seu desempenho em tarefas de regressão e classificação: *Random Forest*, *Support Vector Regression*, *Multilayer Perceptron*, *Long Short-Term Memory* e *Convolutional Neural Networks*. Cada um apresenta particularidades que os tornam mais adequados a diferentes tipos de dados e objetivos, como descrito nas subseções a seguir.

2.2.1. *Random Forest* (RF)

O *Random Forest* é um modelo baseado em uma combinação de várias árvores de decisão, que trabalham em conjunto para aumentar a precisão das previsões. Por meio do método de bagging (*bootstrap aggregating*), cada árvore é treinada com um subconjunto aleatório dos dados, o que contribui para a redução do sobreajuste (*overfitting*) e melhorar a capacidade de generalização. Além disso, sua estrutura baseada em árvores lida bem com dados não lineares, com valores ausentes ou de diferentes escalas, por conta disso é amplamente empregado tanto em tarefas de regressão quanto de classificação [Breiman 2001]

2.2.2 *Support Vector Regression* (SVR)

O SVR é uma extensão das Máquinas de Vetores de Suporte (SVMs), adaptada para tarefas de regressão. O algoritmo projeta os dados para um espaço de maior dimensionalidade por meio de funções kernel e ajusta uma função linear capaz de prever os valores dentro de uma margem de tolerância (epsilon). O SVR apresenta bom desempenho em cenários com relações não lineares e alta dimensionalidade, além de apresentar boa resistência a outliers. No entanto, seu desempenho é fortemente influenciado pela escolha do kernel e dos hiperparâmetros, como o parâmetro de penalidade (C) e a largura da margem (epsilon) [Drucker et al. 1997].

2.2.3. Multilayer Perceptron (MLP)

O MLP é uma rede neural composta por múltiplas camadas de neurônios interligados, projetada para aprender relações complexas e não lineares nos dados. Ele utiliza funções de ativação como ReLU ou sigmoide, e realiza o ajuste dos pesos por meio de retropropagação (*backpropagation*). Embora bastante versátil, o MLP exige uma quantidade significativa de dados para treinar de forma eficaz e requer atenção ao ajuste de hiperparâmetros para evitar problemas como sobreajuste ou dificuldades de convergência [Kumar 2023].

2.2.4. Long Short-Term Memory (LSTM)

As redes LSTM são uma evolução das redes neurais recorrentes (RNN), desenvolvidas para lidar com dados sequenciais e capturar relações temporais de longo prazo. Elas utilizam células de memória com estruturas de controle chamadas "portas" (gates), que funcionam como filtros inteligentes, regulando o que deve ser armazenado, esquecido ou atualizado a cada etapa do processamento. Essa característica permite que as LSTM se destaquem em tarefas como previsão de séries temporais, análise de linguagem natural e reconhecimento de padrões em sequências. Em contrapartida, seu treinamento costuma ser mais custoso computacionalmente e exigem maior volume de dados para alcançar resultados satisfatórios [Hochreiter e Schmidhuber 1997].

2.2.5. Convolutional Neural Networks (CNN)

As CNNs são redes neurais profundas que foram desenvolvidas inicialmente para tarefas envolvendo visão computacional, mas seu uso tem se expandido de forma satisfatória para outros tipos de dados, incluindo séries temporais. Sua estrutura é composta por camadas que imitam o funcionamento do sistema visual humano: as camadas convolucionais identificam padrões locais nos dados, as camadas de pooling ajudam a reduzir a complexidade, e as camadas densas finalizam o processo, realizando a predição. A vantagem principal das CNNs é sua capacidade de reconhecer estruturas hierárquicas e padrões nos dados. Apesar de serem poderosas, a demanda de recursos computacionais é alta, além de atenção no ajuste da arquitetura e dos hiperparâmetros [Lecun et al. 2015].

2.3. Métricas de Avaliação

Foram utilizadas três métricas principais para avaliação dos modelos: o Coeficiente de Determinação, o Erro Quadrático Médio e o tempo de execução. O conjunto dessas métricas permite uma análise equilibrada entre acurácia e viabilidade prática dos modelos.

2.3.1. Coeficiente de Determinação (R^2)

O Coeficiente de Determinação indica o quanto o modelo consegue explicar a variação dos dados, medindo a proporção da variância da variável dependente que é explicada pelas variáveis independentes no modelo. Essa métrica é fundamental para avaliar o grau de ajuste do modelo aos dados observados. O valor de R^2 varia entre 0 e 1, sendo que valores mais próximos de 1 indicam um melhor desempenho preditivo [Kuhn e Johnson 2013]. A fórmula para calcular o R^2 é:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - u_i)^2}{\sum_{i=1}^n (y_i - x)^2} \quad (1)$$

Onde:

y = representa o valor real observado,

u = é o valor previsto pelo modelo,

x = é a média dos valores reais,

n = é o número total das observações.

2.3.2. Erro Quadrático Médio (MSE)

O MSE mede o tamanho médio dos erros de previsão, sendo especialmente importante porque penaliza com mais intensidade os erros mais distantes da realidade, o que é fundamental em aplicações sensíveis como a previsão de casos de dengue [Géron 2019] e quanto mais próximo de 1, melhor é o modelo. A fórmula do MSE é:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - u_i)^2 \quad (2)$$

Onde:

y = representa o valor real observado,

u = é o valor previsto pelo modelo,

n = é o número total das observações.

2.3.3. Tempo de execução

O tempo de execução corresponde à duração necessária para o treinamento dos modelos para que esteja apto para fazer previsões. Essa métrica é crucial na avaliação da viabilidade prática dos algoritmos, principalmente em ambientes que exigem resposta em tempo quase real ou operam com recursos computacionais limitados.

Modelos complexos como redes neurais convolucionais (CNN) ou redes recorrentes (LSTM) geralmente oferecem maior acurácia, mas podem demandar mais tempo de processamento. Assim, medir o tempo de execução permite avaliar o custo computacional associado à precisão dos modelos, conforme argumentado por Goodfellow, Bengio e Courville (2016).

3. Metodologia

A metodologia adotada compreendeu uma sequência estruturada de etapas que iniciou com a coleta de dados seguida pela análise exploratória dos dados, pré-processamento de anomalias encontradas, divisão de dados de treino e teste, otimização dos hiperparâmetros e finalmente o treinamento e avaliação dos modelos como mostrado no fluxograma a seguir:

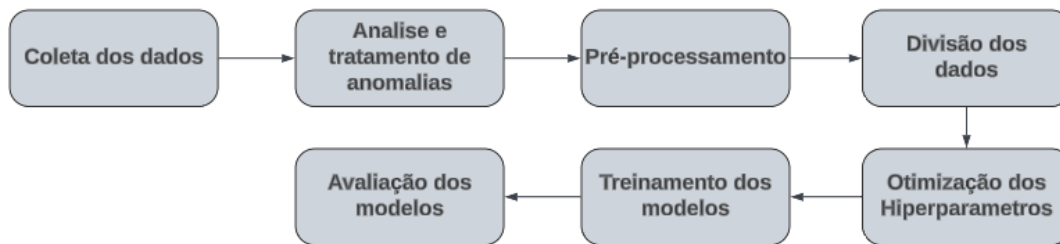


Imagem 1 - fluxograma dos processos.

3.1. Coleta de Dados

O conjunto de dados utilizado foi composto por informações epidemiológicas e climáticas provenientes de fontes públicas oficiais do período compreendido entre os anos de 2010 até 2024. O número de casos confirmados de dengue foi obtido do Sistema de Informação de Agravos de Notificação (SINAN), disponibilizado pelo Ministério da Saúde por meio da plataforma DataSUS, garantindo a confiabilidade dos registros. Os dados meteorológicos (temperatura média, precipitação acumulada e umidade relativa do ar) foram coletados da estação meteorológica da Universidade Federal Rural do Semi-Árido (UFERSA), o EMA, assegurando precisão nas medições ambientais.

3.1.1 Base de Dados

As variáveis da base de dados são estruturadas com o foco na modelagem preditiva da dengue, organizadas temporalmente por mês. O atributo *date* é o marcador temporal de cada observação. As variáveis climáticas: *rainfall_index* (índice de precipitação em mm), *air_humidity* (umidade relativa do ar em g/m³) e *mean_temperature* (temperatura média em °C); são reconhecidas na literatura como fatores de grande influência no ciclo de vida e a proliferação do mosquito *Aedes aegypti*, afetando desde a formação de criadouros até a taxa de desenvolvimento larval e a sobrevivência do vetor [Beserra et al. 2009]. Por fim, *dengue_cases* (número de casos do mês), fornece o cenário de infestação da doença ao longo do tempo. A combinação desses atributos possibilita uma modelagem assertiva da dinâmica de transmissão da dengue. A estrutura completa da base de dados pode ser visualizada a seguir na Tabela 1.

Nome da variável	Descrição
date	Referência temporal da coleta dos dados.
rainfall_index	Índice de chuva registrada no mês, em milímetros (mm).
air_humidity	Umidade relativa média do mês, em porcentagem (%).
mean_temperature	Média de temperatura do ar mensal, em graus Celsius (°C).
dengue_cases	Número de casos de dengue registrados no mês.

30days_cases	Número de casos de dengue registrados nos 30 dias anteriores ao mês atual.
60days_cases	Número de casos de dengue registrados nos 60 dias anteriores ao mês atual.

Tabela 1 - atributos da base de dados.

3.2. Análise e Pré-processamento dos dados

3.2.1. Redução das amostras

O conjunto de dados original estava no formato diário e foi submetido a um processo de conversão para formatação mensal. Para isso, os valores das variáveis temperatura média, umidade relativa do ar e índice de precipitação, foram agregados por média aritmética mensal, enquanto os casos de dengue foram consolidados por soma simples mensal. Esta transformação reduziu significativamente a dimensionalidade do conjunto de dados de aproximadamente 5.400 registros para cada ano analisado para 180 registros totais, sem perder informações relevantes para análise epidemiológica e diminuindo a complexidade computacional.

3.2.2. Tratamento de outliers

Inicialmente, foi realizada uma análise exploratória dos dados de todos os atributos por meio de visualizações em boxplots, com o objetivo de identificar valores destoantes. Sendo identificado valores anômalos (*outliers*), aplicou-se o método estatístico do Intervalo Interquartil (IQR), considerando como limiares 1,5 vezes o IQR abaixo do primeiro quartil e acima do terceiro quartil definindo um limite inferior e superior para os dados coletados. As etapas do procedimento podem ser visualizadas na Imagem 2, onde acontece a identificação inicial das anomalias e o resultado do tratamento dos dados inconsistentes, assegurando maior confiabilidade para as etapas posteriores de modelagem.

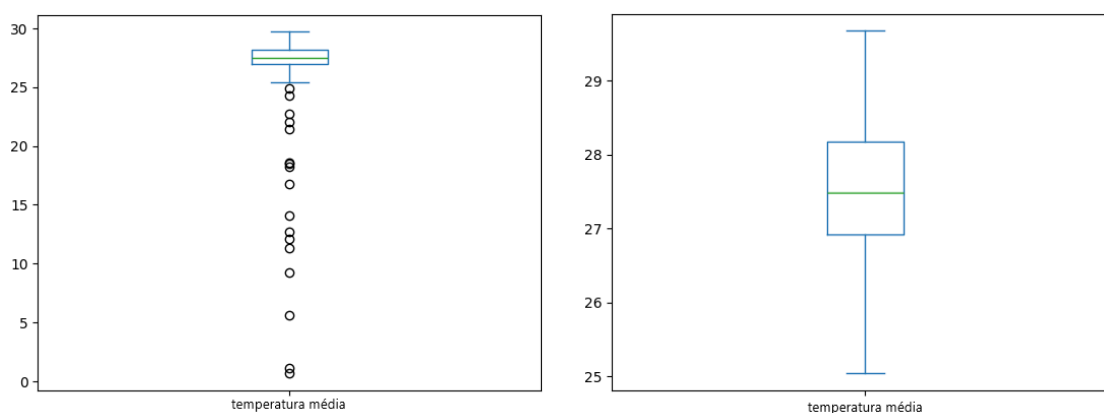


Imagem 2 - boxplot da temperatura média antes e depois do IQR.

3.3. Divisão de dados de treino e teste

Para a avaliação dos modelos, os dados foram divididos em dois subconjuntos: 70% foram utilizados para o treinamento dos algoritmos e os 30% restantes reservados para teste. Essa proporção garantiu que os modelos aprendessem de forma eficaz, ao mesmo tempo avaliados de maneira satisfatória pelos dados não vistos, assegurando a generalização dos resultados.

3.4. Parâmetros de execução dos algoritmos

Foi utilizada a técnica de GridSearchCV, que é um classe da biblioteca Scikit-Learn, para otimizar os hiperparâmetros de todos os algoritmos avaliados. Esta abordagem avalia combinações de parâmetros de forma exaustiva através de validação cruzada, para identificar a combinação de parâmetros que maximizam o desempenho preditivo. Nas redes neurais LSTM e CNN, o GridSearchCV foi aplicado para otimizar o número de camadas, filtros convolucionais e taxa de dropout. Na Tabela 2 são apresentados os resultados obtidos para cada algoritmo:

Random Forest	max_depth = 10, min_samples_leaf = 4, min_samples_split = 5, n_estimators = 600, n_jobs = -1
Support Vector Regression	C = 5, coef0 = 0.01, epsilon = 2.0, gamma = 'auto', kernel = 'linear'
Multi Layer Perceptron	activation = 'logistic', learning_rate_init = 0.01, max_iter = 5000
Long Short-Term Memory	Neurons: 50, Activation: relu, Recurrent Activation: tanh, Dropout: 0.0, Batch size: 12
Convolutional Neural Network	Neurons: 25, Activation: relu, Recurrent Activation: sigmoid, Dropout: 0.1, Batch Size: 12

Tabela 2 - Parâmetros obtidos para o treinamento de cada modelo.²

3.5. Treinamento dos modelos

Após a definição dos melhores hiperparâmetros por meio do GridSearchCV, os algoritmos foram treinados utilizando essas configurações otimizadas. Dessa forma, cada modelo foi analisado em sua configuração mais eficiente, permitindo uma comparação justa entre as diferentes abordagens.

3.6. Avaliação dos modelos

Na etapa de avaliação, utilizou-se a validação cruzada com o objetivo de reduzir o risco de overfitting e obter estimativas mais confiáveis da performance dos modelos. Essa abordagem consiste em dividir o conjunto de treinamento em múltiplas partes (folds), utilizando algumas para treinar e outras para validar o modelo, de forma alternada.

² [GridSearchCV — scikit-learn 1.6.1 documentation](https://scikit-learn.org/stable/modules/grid_search.html)

3.7. Ambiente de execução

Os experimentos foram realizados em uma máquina com CPU Ryzen 5 4600G com 16,0 GB de memória RAM DDR4, GPU NVIDIA RTX 2060 Super (6GB), utilizando a IDE Visual Studio Code visando garantir compatibilidade nos tempos de execução reportados.

4. Resultados obtidos

Após a execução dos algoritmos, foram constatados os seguintes resultados presentes na Tabela 3:

Algoritmo	R ² Score	MSE Score	Tempo de execução
Random Forest	0.5950	71.649	2m 17s
Support Vector Regression	0.6621	65.450	30.4s
Multi Layer Perceptron	0.6268	68.779	4m 29.3s
Long Short-Term Memory	0.7942	61.349	5m 35.2s
Convolutional Neural Network	0.7956	61.144	19m 0.4s

Tabela 3 - Resultado das métricas.

Com o objetivo de verificar se as diferenças de desempenho entre os algoritmos testados são estatisticamente significativas, foi aplicado o Teste de Friedman.

4.1. Teste de Friedman

O teste de Friedman é um método não paramétrico amplamente utilizado para comparar múltiplos algoritmos em diferentes conjuntos de dados ou métricas, especialmente útil quando não se pode assumir a normalidade das distribuições. Esse teste ordena o desempenho dos algoritmos por rankings — atribuindo 1 ao melhor desempenho, 2 ao segundo melhor, e assim sucessivamente — para cada cenário avaliado. Posteriormente, calcula-se a média dos rankings e avalia-se a hipótese nula de que todos os algoritmos possuem desempenhos equivalentes. A verificação dessa hipótese é feita por meio de um valor estatístico que é comparado à distribuição qui-quadrado. Se o valor-p (p-value) resultante for inferior ao nível de significância adotado (geralmente 0,05), conclui-se que há diferença estatisticamente significativa entre os algoritmos.

No presente estudo, em relação à métrica de R², o teste de Friedman resultou em $p = 0,0008$, indicando diferenças estatisticamente significativas entre os modelos. Como etapa subsequente, foi aplicado o pós-teste de Nemenyi para identificar quais algoritmos apresentaram diferenças relevantes de desempenho.

4.1.2 Pós-teste de Nemenyi

Caso o teste de Friedman indique diferenças significativas, aplica-se o teste de Nemenyi, que realiza comparações par a par entre os algoritmos. Esse teste determina quais pares de modelos

apresentam desempenhos estatisticamente diferentes, considerando um nível de significância pré-definido [Demasar 2006].

Após a execução do teste foram obtidos os seguintes resultados apresentados na Imagem 3.

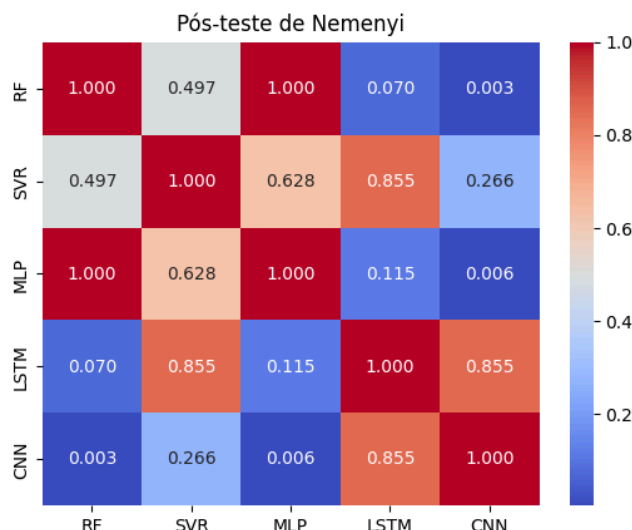


Imagem 3 - Pós-teste de Nemenyi.

A análise estatística destacou o modelo CNN, que apresentou desempenho significativamente superior ao Random Forest e ao MLP. Embora a LSTM e o SVR também tenham obtido bons resultados, suas diferenças em relação à CNN não foram estatisticamente significativas. Isso reforça a eficácia dos modelos de deep learning no problema proposto, especialmente da CNN, enquanto os métodos tradicionais apresentaram desempenho inferior.

4.3 Avaliação dos resultados

A avaliação comparativa em termos das métricas de R^2 e tempo de execução permite analisar qual algoritmo é adequado a diferentes cenários reais, considerando situações em que a precisão das previsões mais significativa e outras em que a rapidez de processamento é prioridade. Apresentando os modelos por ordem de eficácia crescente, o *Random Forest* apresentou o menor poder de predição, com R^2 de 0,5950 e o maior erro de todos, sendo o MSE de 71,649, mostrando não ser uma boa alternativa para este cenário, mesmo com um tempo de execução relativamente baixo (2 minutos e 17 segundos).

O *Multilayer Perceptron* apresentou o pior equilíbrio entre desempenho e tempo de processamento. Com R^2 de 0,6268 e MSE de 68,779, sua eficácia ficou abaixo tanto de modelos clássicos otimizados, como o SVR (que obteve R^2 superior e menor MSE), quanto das arquiteturas de *Deep Learning*, além disso, apresentou um tempo de execução quase duas vezes maior que o *Random Forest*, com resultados muito semelhantes.

O *Support Vector Regression* apresentou um $R^2 = 0,6621$ e $MSE = 65,450$, sendo o mais eficiente em tempo de processamento (30,4 segundos), o que indica um bom equilíbrio entre precisão e custo computacional.

Por fim, os modelos de *Deep Learning* se destacaram pela alta acurácia, corroborando com o teste de Friedman realizado anteriormente: a LSTM alcançou R^2 de 0,7942 e MSE de

61,349, enquanto a CNN superou ligeiramente com R^2 de 0,7956 e MSE de 61,144, confirmando sua eficácia na detecção de padrões temporais. No entanto, o custo computacional foi muito elevado, no caso da CNN, que levou 19 minutos e 0,4 segundos para executar, ou seja, quase quatro vezes mais do que a LSTM (5 minutos e 35,2 segundos), com uma diferença percentual de apenas 0,18% no R^2 e 0,33% no MSE.

A relação entre R^2 e tempo de execução (em segundos) é melhor visualizada no Gráfico 1 a seguir:

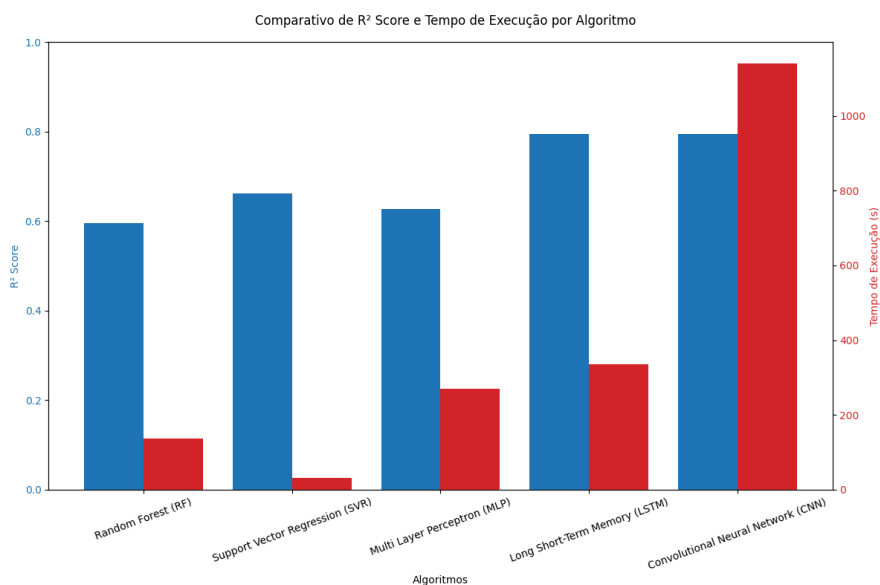


Gráfico 1 - Relação R^2 e tempo de execução.

O LSTM obteve um dos melhores resultados de acurácia ($R^2 \approx 0,79$) com um tempo de execução relativamente baixo (cerca de 335 segundos), mostrando excelente equilíbrio entre precisão e eficiência computacional.

5. Conclusão

Os resultados mostram que, apesar das redes neurais profundas (LSTM e CNN) oferecerem a melhor precisão, o SVR pode ser a melhor alternativa quando o tempo de processamento é um fator crítico. A LSTM se destacou por alcançar um dos maiores R^2 com um custo computacional significativamente menor do que o da CNN, que demandou quase quatro vezes mais tempo com a diferença no poder de predição irrelevante nesse cenário. O RF e o SVR apresentaram desempenho satisfatório, porém inferior aos demais modelos em termos de métricas de erro e capacidade explicativa. Assim, a escolha do algoritmo ideal dependerá do equilíbrio entre a precisão desejada e os recursos computacionais disponíveis.

Além disso, os modelos avaliados possuem potencial para aplicação prática em sistemas de apoio à decisão voltados à saúde pública, como a construção de painéis interativos para visualização de dados. Esses sistemas podem auxiliar gestores na identificação antecipada de regiões de risco, contribuindo para a alocação mais eficiente de recursos e a implementação de medidas preventivas. Um exemplo concreto dessa aplicação foi o desenvolvimento de um painel de visualização baseado nas previsões geradas, permitindo uma análise temporal e geográfica mais acessível e dinâmica.

Como perspectivas para trabalhos futuros, sugere-se o uso de dados com maior

granularidade espacial (como por bairro ou zona), a incorporação de informações sobre mobilidade humana, e a experimentação com modelos híbridos, a fim de aumentar ainda mais a acurácia e a capacidade de adaptação das previsões. Adicionalmente, a automação da atualização dos dados e a integração com plataformas públicas de dados abertos podem tornar o sistema mais ágil e aplicável a contextos reais de vigilância epidemiológica.

Referências

- ALEIXO, Helena et al. Modelo de aprendizado de máquina pode ser utilizado para prever surtos de dengue. Agência FAPESP, São Paulo, 08 abr. 2022. Disponível em: <https://agencia.fapesp.br/modelo-de-aprendizado-de-maquina-pode-ser-utilizado-para-prever-surtos-de-dengue/39217>.
- Beserra, E. B. et al. (2009). Efeito da temperatura sobre o ciclo de vida, exigências térmicas e estimativas do número de gerações anuais de *Aedes aegypti* (Diptera, Culicidae). *Iheringia. Série Zoologia*, 99(2):142–148. Disponível em: <https://www.scielo.br/j/isz/a/xrWcVLDyrm9dBMKP4JJXXkN>.
- Bhatt, S. et al. (2013). The global distribution and burden of dengue. *Nature*, 496:504–507. DOI: 10.1038/nature12060.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Brasil. Ministério da Saúde. (2023). Monitoramento dos casos de arboviroses urbanas transmitidas pelo *Aedes* (dengue, chikungunya e Zika), semanas epidemiológicas 1 a 52 – 2023. Boletim Epidemiológico, Brasília. Disponível em: <https://www.gov.br/saude/pt-br>.
- Demasar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, v. 7, p. 1–30, 2006.
- Drucker, H. et al. (1997). Support vector regression machines. In: *Advances in Neural Information Processing Systems*, p. 155–161.
- Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2. ed. Sebastopol: O'Reilly Media.
- Goodfellow, I.; Bengio, Y.; Courville, A. (2016). *Deep Learning*. Cambridge: MIT Press.
- Hochreiter, S.; Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Kumar, R. (2023). *Perceptrons Multicamadas em Aprendizado de Máquina*. DataCamp. Disponível em: <https://www.datacamp.com/pt/tutorial/multilayer-perceptrons-in-machine-learning>.
- Kuhn, M.; Johnson, K. (2013). *Applied Predictive Modeling*. New York: Springer. DOI: 10.1007/978-1-4614-6849-3.
- Lecun, Y.; Bengio, Y.; Hinton, G. (2015). Deep learning. *Nature*, 521:436–444.
- Morin, C. W.; Comrie, A. C.; Ernst, K. (2013). Climate and dengue transmission: evidence and implications. *Environmental Health Perspectives*, 121(11–12):1264–1272. DOI: 10.1289/ehp.1306556.
- Organização Mundial da Saúde. (2017). *Global vector control response 2017–2030*. Genebra. Disponível em: <https://www.who.int/publications/i/item/9789241512978>.
- Organização Mundial da Saúde. (2022). *Dengue and severe dengue*. Genebra. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue>.