UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JOÃO MAIERON MARTINS

Explorando Dados Climáticos e do Google Trends na Predição de Casos de Dengue no Brasil: Uma Abordagem com Aprendizado de Máquina

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Profª. Drª. Mariana Recamonde Mendoza

Porto Alegre
2024

AGRADECIMENTOS

Agradeço à minha orientadora Mariana Recamonde Mendoza pela atenção, paciência, dedicação e conhecimento que tornaram este trabalho possível. Agradeço a meus pais pelo carinho, pelo amor e segurança que me permitiram chegar até aqui. Aos amigos que me acompanharam no percurso, em especial ao Diego, Arthur e Bruno, pelo companheirismo que tornou essa graduação mais leve. E a todos os excelentes professores que contribuíram para a minha formação.

RESUMO

A dengue é uma arbovirose que representa grande perigo à saúde pública no Brasil, onde anualmente são registrados números altíssimos de casos, e surtos têm se tornado cada vez mais recorrentes. Nos últimos anos, técnicas de aprendizado de máquina têm se mostrado muito úteis em esforços de monitoramento e contenção dessa doença. No entanto, esforços do tipo são bastante recentes e há muitos caminhos ainda não explorados. Este trabalho consiste na montagem de modelos preditivos de casos de dengue para todas as unidades federativas do Brasil, com uso das técnicas *Random Forest* (RF), *Support Vector Regressor* (SVR) e *Long Short-term Memory* (LSTM), treinados sobre séries temporais que englobam dados climáticos e números de buscas na Internet. Além de previsões por modelos individuais, foi testada uma modalidade *ensemble* por macrorregião, a fim de explorar a fundo a possibilidade de previsões nesse nível no país. Realizados os experimentos, observou-se grande variação de performance sobre cada estado, indicando que há entre eles fortes diferenças na qualidade dos dados ou nas formas como se manifesta a doença, o que torna alguns deles mais favoráveis a predições desse tipo. Percebeu-se leve queda de performance com o uso de *ensemble*, o que reforçou esse entendimento. Os modelos conseguiram, sobre uma parcela dos estados, captar bem as tendências temporais, ainda que com discrepâncias nos números exatos de casos. Também comprovaram-se a utilidade de dados do Google Trends e a boa adequação de SVR e LSTM para pesquisas deste tipo. Embora os modelos construídos não sejam precisos o bastante para o monitoramento da dengue, espera-se que as descobertas feitas possam direcionar esforços futuros, com foco nos locais e nas técnicas mais promissoras, para construir modelos mais eficientes.

Palavras-chave: Dengue. Séries temporais. Aprendizado de máquina. LSTM.

Exploring Climate and Google Trends Data to Predict Dengue Cases in Brazil: A Machine Learning Approach

ABSTRACT

Dengue is an arbovirus that poses a significant threat to public health in Brazil, where annually, extremely high numbers of cases are reported, and outbreaks have become increasingly recurrent. In recent years, machine learning techniques have proven highly useful in monitoring and controlling this disease. However, such efforts are relatively recent, and there are many unexplored avenues. This work involves the construction of predictive models for dengue cases in all Brazilian states, using the Random Forest, Support Vector Regressor, and Long Short-term Memory techniques. These models are trained on time series data that encompass climatic information and Internet search volumes. In addition to predictions by individual models, a regional ensemble approach was tested to thoroughly explore the possibility of forecasting at this level in the country. Following the experiments, significant performance variations were observed among states, indicating strong differences in data quality or the manifestation of the disease, making some states more favorable to such predictions. A slight drop in performance with the use of ensembles reinforced this understanding. The models were able to capture temporal trends well in some states, although with discrepancies in the exact numbers of cases. The utility of Google Trends data for such research was also confirmed. While the constructed models may not be accurate enough for dengue monitoring, it is hoped that the findings can guide future efforts, focusing on the most promising locations and techniques to build more efficient models.

Keywords: Dengue. Time series. Machine learning. LSTM.

LISTA DE FIGURAS

Figura 2.1 Procedimento em aprendizado supervisionado 15
Figura 2.2 Estratégias para séries temporais 16
Figura 2.3 Diagrama do método RF 19
Figura 2.4 Diagrama do método SVR 21
Figura 2.5 Arquitetura interna de uma célula de LSTM. 23

Figura 4.1 Fluxograma da montagem dos *datasets* 28
Figura 4.2 Visualização da versão final dos *datasets* 29
Figura 4.3 Casos de dengue no Brasil durante o período em estudo 30
Figura 4.4 Histórico de estações meteorológicas ativas por estado 32
Figura 4.5 Exemplo do *dataset* climático típico. Dados recolhidos para o estado da Bahia. 33
Figura 4.6 Comparativos entre índices do Google Trends e casos de dengue em diferentes estados, com os dados normalizados. 34
Figura 4.7 Fluxograma da montagem e treinamento dos modelos. Processo repetido para cada um dos *datasets*. 35
Figura 4.8 Estrutura final dos modelos de LSTM 38

Figura 5.1 Boxplots para MAE, MAPE e R^2, por método. As linhas verdes tracejadas apresentam as médias; as linhas laranjas, as medianas. 42
Figura 5.2 Distribuição de casos para AP, PA e SP. Em azul, o período de 52 semanas separado para teste. 44
Figura 5.3 Previsões para AP, PA e SP: exemplos de melhores resultados. Em azul, os números reais; em laranja, as predições. 44
Figura 5.4 Distribuição de casos para AM, BA e MT. Em azul, o período de 52 semanas separado para teste. 45
Figura 5.5 Previsões para AM, BA e MT: exemplos de resultados bons. Em azul, os números reais; em laranja, as predições. 45
Figura 5.6 Distribuição de casos para ES, RR e RS. Em azul, o período de 52 semanas separado para teste. 46
Figura 5.7 Previsões para ES, RR e RS: exemplos de resultados ruins. Em azul, os números reais; em laranja, as predições. 46
Figura 5.8 Previsões por LSTM para MG, PR e SE. Na coluna da esquerda, as previsões contrapostas ao todo; na da direita, um *zoom* no período de teste. 47
Figura 5.9 Previsões por LSTM para AC, GO e RO. Na coluna da esquerda, as previsões contrapostas ao todo; na da direita, um *zoom* no período de teste. 48
Figura 5.10 Métricas para o teste sem a utilização de Google Trends. As linhas verdes tracejadas apresentam as médias; as linhas laranjas, as medianas. 49
Figura 5.11 Exemplo de aplicação do *ensemble* por macrorregião 49
Figura 5.12 Distribuição das métricas do *ensemble* por macrorregião. As linhas verdes tracejadas apresentam as médias; as linhas laranjas, as medianas. 50
Figura 5.13 Comparativo de previsões por *ensemble* para RJ e PE, com uso de RF e SVR. Na coluna da esquerda, as previsões originais; na da direita, aquelas feitas por *ensemble* 51
Figura 5.14 Comparativo de previsões por *ensemble* para RN, AC e RO, com uso dos três métodos. Na coluna da esquerda, as previsões originais; na da direita, aquelas feitas por *ensemble* 52

Figura 5.15 Comparativo de previsões por *ensemble* para AL, AM e DF, com uso de LSTM. Na coluna da esquerda, as previsões originais; na da direita, aquelas feitas por *ensemble*. 53

LISTA DE TABELAS

Tabela 4.1 Hiperparâmetros para SVR e regressor por RF 37

Tabela 5.1 Valores de MAE para todos os modelos. Destacam-se os cinco melhores colocados, por método. 40
Tabela 5.2 Valores de R^2 e MAPE para todos os modelos. Destacam-se os melhores colocados dentro de cada algoritmo, por métrica. 41

LISTA DE ABREVIATURAS E SIGLAS

INMET	Instituto Nacional de Meteorologia
LSTM	Long Short-Term Memory
ML	Machine Learning
MAE	Mean absolute error
MSE	Mean squared error
RF	Random Forest
RNR	Rede neural recorrente
SINAN	Sistema de Informação de Agravos de Notificação
SVM	Support Vector Machine
SVR	Support Vector Regressor

SUMÁRIO

1 INTRODUÇÃO **11**
2 REFERENCIAL TEÓRICO **13**
2.1 Conceitos gerais de *Machine Learning* **13**
2.1.1 Aprendizado supervisionado 13
2.1.2 Procedimentos 14
2.1.3 Séries Temporais 15
2.1.4 Métricas de avaliação 17
2.2 Métodos selecionados **18**
2.2.1 *Random Forest* 18
2.2.2 *Support Vector Regressor* 20
2.2.3 *Long Short-Term Memory* 22
3 TRABALHOS RELACIONADOS **24**
3.1 Uso de ML em predições de série temporal **24**
3.2 Uso de ML em monitoramento de dengue **25**
4 METODOLOGIA **28**
4.1 Coleta e pré-processamento dos dados **28**
4.1.1 Histórico de dengue no Brasil 29
4.1.2 Dados climáticos 30
4.1.3 Números do Google Trends 32
4.2 Montagem e treinamento dos modelos **34**
4.2.1 Preparativos para o treinamento 34
4.2.2 Otimização de hiperparâmetros 35
4.2.3 Estrutura final dos modelos 36
5 RESULTADOS **39**
5.1 Previsões individuais por estado **39**
5.1.1 Comparativos entre os algoritmos 42
5.1.2 Estudos de caso - padrões gerais 43
5.1.3 Estudos de caso - LSTM 46
5.1.4 Impacto dos dados do Google Trends 48
5.2 *Ensemble* por macrorregião **49**
6 CONCLUSÃO **54**
REFERÊNCIAS **56**

1 INTRODUÇÃO

A dengue é uma doença viral endêmica no Brasil, onde as condições climáticas favorecem a proliferação dos insetos vetores, os mosquitos do gênero Aedes. Dada a dimensão da doença no país, ela foi inserida entre aquelas de notificação compulsória ao sistema de saúde público. O recolhimento e tratamento dessas notificações é feito através do SINAN (Sistema de Informação de Agravos de Notificação), base de dados epidemiológicos que é constantemente alimentada para fins de monitoramento. A base foi iniciada em 1993 e, no caso da dengue em específico, guarda dados que remontam ao ano 2000[1].

Leituras estatísticas feitas sobre o período coberto por esse sistema observam um aumento de frequência e incidência de surtos de dengue, com registro de epidemias de mais de 1.000.000 de casos no país em 2013, 2015, 2016, e 2019 (Junior et al., 2022). Embora tenha havido diminuição no registro de casos no ano de 2020, isso é atribuído a uma subnotificação em função da situação anômala da pandemia de COVID-19, com os anos posteriores voltando a exibir o mesmo padrão crescente de antes (Neto et al., 2023).

Uma das estratégias tomadas ao longo do tempo como resposta a esse avanço da patologia foi o uso de técnicas de aprendizado de máquina (do inglês, *Machine Learning*, abreviado como ML) para detectar padrões de manifestação, o que abriria espaço para a tomada de medidas preventivas. Esforços nesse sentido vêm sendo tomados em diversos países, com diferentes resultados, devido às peculiaridades do avanço do vírus em cada local. No Brasil, a existência do SINAN oferece o apoio perfeito para esse tipo de abordagem.

Este Trabalho de Graduação busca somar-se a esses esforços de monitoramento, construindo e analisando a capacidade de diferentes modelos de ML para prever a incidência de casos da doença em escala estadual no Brasil. Os modelos são treinados sobre números do SINAN em conjunto com dados climáticos recolhidos pelo INMET (Instituto Nacional de Meteorologia). Dados desse tipo são obrigatórios para o estudo da dengue, visto que as taxas de reprodução dos mosquitos vetores estão fortemente relacionadas a determinadas condições de temperatura e precipitação (Pliego; Velázquez-Castro; Collar, 2017).

Os conjuntos de dados para o trabalho também incluem quantidades de pesquisas envolvendo o nome da doença no motor de buscas Google, recolhidas através do portal

[1] Informações de <http://portalsinan.saude.gov.br>

Google Trends. Pressupõe-se que em períodos de maior manifestação da doença, haverá maior interesse da população em buscar informações a seu respeito, o que justifica a inserção desses dados no estudo. Por outro lado, a imprecisão inerente a esses números e à maneira como são disponibilizados coloca certa dúvida sobre sua confiabilidade (Marques-Toledo et al., 2017). Assim, dentro da proposta, faz-se também uma rápida análise do impacto desses dados sobre a performance dos modelos escolhidos.

Estudos anteriores que atacaram o mesmo problema demonstraram ser factíveis previsões a nível de cidade no Brasil, contanto que haja um conjunto de dados suficientemente detalhado. Foi buscando explorar a possibilidade de previsões em maior escala que, neste trabalho, optou-se pelo agrupamento de dados a nível estadual. Por fim, levando em conta padrões de movimentação espacial da doença, é construído um método de previsões *ensemble* por macrorregião do país.

Em resumo, este trabalho objetiva analisar se é possível realizar previsões de casos de dengue no país, por estado, e se faz sentido o uso de dados do Google Trends no auxílio dessas previsões. Ao final, os resultados obtidos demonstram que existe sim possibilidade de realizar previsões sob esses parâmetros, porém com ressalvas quanto à qualidade dos dados referentes a um subconjunto dos estados. Também é demonstrado que números do Trends são valiosos para experimentos do tipo.

O trabalho está estruturado da seguinte forma: no 2º capítulo são descritos os conceitos básicos necessários à compreensão da pesquisa; no 3º, faz-se um apanhado de trabalhos relevantes na área, que serviram como base para este; no 4º, é descrita a metodologia adotada; no 5º são descritos os resultados, e finalmente o 6º traz as considerações finais.

2 REFERENCIAL TEÓRICO

Explicam-se a seguir os conceitos e métodos relevantes para o entendimento deste estudo. O capítulo inicia descrevendo brevemente alguns fundamentos relevantes de *Machine Learning*, e na sequência detalha os métodos escolhidos para a construção dos preditores.

2.1 Conceitos gerais de *Machine Learning*

Aprendizado de máquina (que daqui em diante será referido pela abreviatura ML) é o campo da inteligência artificial que lida com o desenvolvimento de algoritmos capazes de identificar padrões em conjuntos de dados e com base nisso tomar decisões, tudo de forma autônoma (Alpaydin, 2020). Em termos mais simples, algoritmos desse tipo aprendem através de exemplos e assim conseguem interpretar informações novas. Possuem incontáveis aplicações, como em diagnósticos médicos, processamento de linguagem natural, recomendações personalizadas em redes sociais.

2.1.1 Aprendizado supervisionado

Algoritmos de ML são categorizados de acordo com as tarefas que se propõem a cumprir. Quando objetiva-se montar um modelo capaz de fazer predições ou categorizar dados, fala-se em aprendizado supervisionado. Modelos desse tipo são treinados sobre conjuntos de dados em que cada instância contém um atributo alvo, também chamado variável dependente. Durante o seu treinamento, o algoritmo busca entender a relação entre o alvo e os demais atributos de cada instância, também chamados de variáveis independentes. Quando bem treinado, espera-se que o algoritmo consiga mapear as variáveis independentes à dependente.

As tarefas em que se aplica aprendizado supervisionado dividem-se em problemas de classificação ou de regressão. Na primeira modalidade, encaixam-se aqueles problemas em que o objetivo é classificar cada instância do conjunto de treino em alguma categoria específica. Já em regressão, o atributo alvo é um valor numérico.

2.1.2 Procedimentos

Em geral, o desenvolvimento de um algoritmo de ML inicia-se pela coleta e pré-processamento dos dados de entrada, etapa na qual organiza-se esse conjunto para que possa ser corretamente lido. Isso envolve o uso de estratégias para lidar com ruídos e anomalias nos dados, bem como transformá-los para um formato que seja compatível com o algoritmo escolhido (Faceli et al., 2021). Quando um conjunto de dados é alimentado a um algoritmo para que ele os leia, diz-se que está sendo efetuado o seu treinamento.

Em aprendizado supervisionado, os dados pré-processados são divididos em conjuntos de treino e de teste, em proporções definidas manualmente. Usualmente separam-se 70 ou 80% dos dados para treinamento, mas não há regras quanto a isso. O conjunto de treino contém os exemplos para o aprendizado do algoritmo e o de teste servirá para a sua avaliação final.

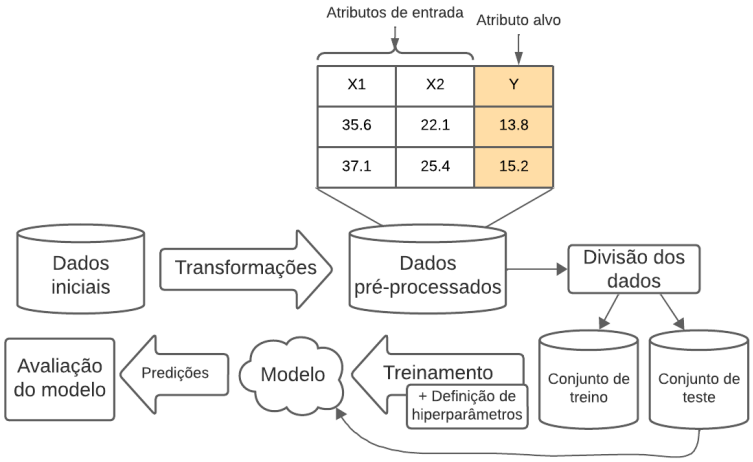
A etapa seguinte é a da montagem da estrutura do algoritmo que será utilizado, incluindo a escolha dos hiperparâmetros, que são configurações externas que afetam o seu comportamento. Os hiperparâmetros podem ser definidos manualmente ou através de algum algoritmo de busca que tenta definir as melhores configurações de acordo com o conjunto de dados de entrada. Finalmente, passa-se ao treinamento do algoritmo em si. A estrutura que resulta desse treinamento é denominada modelo e estará preparada para executar tarefas com base no que aprendeu.

Existem técnicas que auxiliam no processo de montagem, indicando para o desenvolvedor se o modelo que está sendo construído possui boas perspectivas de funcionamento. Uma delas é a validação cruzada, em que os dados são divididos em subconjuntos, e o algoritmo é treinado e testado alternadamente sobre diferentes grupos dessas divisões (Mohri; Rostamizadeh; Talwalkar, 2018). Pode ser utilizada durante o treinamento ou mesmo como avaliação do modelo final.

Efetuado o treinamento, o modelo é avaliado através de leituras feitas por ele sobre o conjunto de teste que foi separado anteriormente. Essas leituras, que no caso de aprendizado supervisionado serão predições, são comparadas aos valores reais e avaliadas de acordo com métricas que o desenvolvedor tenha definido. Uma simplificação de todo o processo pode ser vista na imagem 2.1.

Como últimos conceitos relevantes, mencionam-se *overfitting* e *underfitting*, dois problemas comuns em ML. *Overfitting* acontece quando o modelo ajusta-se demais aos dados de treino, o que faz com que não possua capacidade de generalização e apresente má

Figura 2.1 – Procedimento em aprendizado supervisionado



Fonte: o autor

performance quando apresentado a dados novos. Já *underfitting* manifesta-se em modelos que não conseguiram aprender durante o treinamento. O cuidado durante as etapas de pré-processamento e de definição de hiperparâmetros é crucial para evitar ambos.

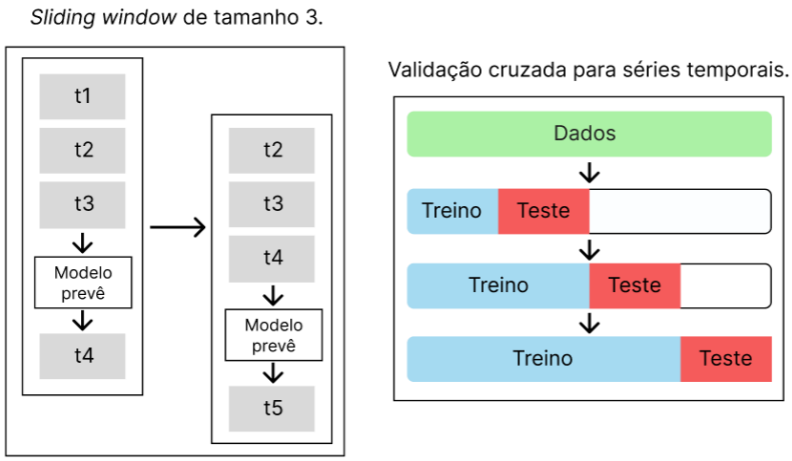
Trabalhos anteriores demonstram que há grande potencial para o uso de ML na área da epidemiologia, sendo esses algoritmos capazes de lidar com a complexidade dos padrões existentes em dados de doenças sazonais como a dengue (Hoyos; Aguilar; Toro, 2021). A escolha dos modelos para este trabalho levou em conta a aplicabilidade para esse tipo de tarefa e também baseou-se na bibliografia do ramo que, embora largamente recente, apresenta uma boa base.

2.1.3 Séries Temporais

No âmbito do aprendizado supervisionado, especial atenção é dada para aqueles conjuntos de dados em que a ordem das instâncias é fundamental para a acurada previsão de valores futuros. São essas as séries temporais, definidas formalmente como conjuntos de dados coletados em ordem cronológica, em que cada observação está associada a um momento específico no tempo. Sequências desse tipo apresentam padrões sazonais de curto ou longo prazo e exigem tratamento diferente das demais categorias de dados (Brockwell; Davis, 2009). Quando do treinamento de um modelo, é imprescindível a manutenção da ordem das entradas, para que esses padrões possam ser capturados.

Há diferentes estratégias para a manutenção dessa ordem na construção de um mo-

Figura 2.2 – Estratégias para séries temporais



Fonte: o autor

delo. Técnicas tradicionais de validação cruzada, por exemplo, não podem ser aplicadas em séries temporais por dividir os dados em subconjuntos tratados como independentes entre si. Para séries temporais, usa-se uma variação da técnica denominada *time-series split*, que garante divisões de forma que os subconjuntos de teste sejam sempre posteriores ao de treino (Lainder; Wolfinger, 2022).

Outra estratégia para o tratamento desses dados é a *sliding window*, em que se define uma janela de observações de tamanho fixo x, que movimenta-se pelo conjunto de dados fazendo com que o valor do atributo alvo a cada tempo t seja sempre previsto com base na leitura das últimas x instâncias (Hota; Handa; Shrivas, 2017). Essa estratégia pode ser bastante efetiva, contanto que a largura da janela seja suficiente para o modelo capturar padrões cíclicos nos dados. A Figura 2.2 contém esquemas simples de *sliding window* e validação cruzada para séries temporais.

Em epidemiologia, a análise de dados históricos organizados em séries temporais é fundamental para o correto estudo do avanço de patologias (Antunes; Cardoso, 2015). Revisão bibliográfica feita por Junior et al. (2022) lista *insights* que foram obtidos de estudos do tipo sobre dengue, como a forte associação da doença com épocas de chuva e o ritmo crescente da ocorrência de surtos a partir de 2010. Através da leitura de um compilado de publicações entre 2009 e 2019 os autores do artigo puderam demonstrar claramente o peso socioeconômico crescente da doença no Brasil.

Séries temporais permitem também a observação de relações complexas entre diferentes atributos em um conjunto de dados ao longo do tempo. Conclusões do tipo são obtidas por Pliego, Velázquez-Castro and Collar (2017) ao apontar o aumento de infec-

ções quando há concomitância de alta precipitação com altas temperaturas. Através de análise histórica de dados climáticos do México, os autores do estudo puderam correlacionar períodos de alta reprodutibilidade do mosquito *Aedes aegypti* com a deflagração de surtos de dengue.

2.1.4 Métricas de avaliação

A avaliação de modelos de ML é feita com base em métricas quantitativas calculadas sobre um comparativo entre os valores preditos e os valores reais contidos no conjunto de teste. A escolha da métrica ideal depende do problema com o qual se está lidando, já que cada uma diz informações diferentes sobre os resultados. Neste trabalho utilizam-se algumas das métricas mais comumente aplicadas em problemas de regressão: o erro médio absoluto, o erro quadrado médio, o erro percentual absoluto médio e o r-quadrado.

Erro absoluto médio (do inglês *mean absolute error*, abreviado como MAE) é a mais intuitiva das métricas listadas, tratando-se da média das diferenças absolutas entre valores reais e previstos. Já o erro quadrado médio (*mean squared error*, ou apenas MSE) faz uma média dos quadrados dessas mesmas diferenças e o erro percentual absoluto médio (*mean absolute percentage error*, ou MAPE) expressa-as em porcentagem. As suas fórmulas estão nas equações (2.1), (2.2) e (2.3), onde n é o número de instâncias de teste e $\hat{y}_i$ e y_i são, respectivamente, os valores previstos e reais para a instância i.

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

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

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{2.3}$$

Embora semelhantes, MAE e MSE possuem uma diferença crucial na maneira como tratam erros de grande valor. Em MSE esses erros são mais severamente punidos, enquanto em MAE todos os erros são tratados igualmente, dessa forma os valores entre elas podem ser radicalmente diferentes em *datasets* que contenham muitos *outliers*, ou pontos anômalos. Neste trabalho, o MSE é separado para uso como função objetivo na construção dos modelos, buscando montar arquiteturas que minimizem erros gritantes. Já MAE e MAPE são utilizados na avaliação dos resultados, a primeira pela sua fácil

compreensão e a segunda por ser útil em comparativos, já que independe da escala das variáveis.

R-quadrado (abreviado para R²) é uma medida estatística que denota o quanto as variações no atributo alvo são explicadas pelo modelo. É expressa em porcentagem, de modo que o valor 1 significa que o modelo explica 100% das flutuações no atributo alvo. No contexto de séries temporais, pode ser entendida como a medida do quanto um modelo conseguiu capturar os padrões temporais e relações subjacentes entre as variáveis da série. Matematicamente, é expresso pela equação (2.4), onde $\hat{y}_i$ são valores previstos, y_i são valores reais e $\bar{y}$ é a média dos valores da variável dependente.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.4}$$

O R² é bastante popular no campo da estatística, possivelmente por ser de fácil interpretação, mas há críticas a sua utilidade em avaliação de preditores, já que essa métrica olha apenas para o comportamento das variáveis em relação uma a outra, sem considerar seus valores (Li, 2017). Isso significa que um modelo que obtenha bom R² pode ainda assim estar errando bastante em relação aos valores previstos em si. Assim, por si só essa métrica não informa sobre a acurácia de um modelo, devendo ser acompanhada de métricas baseadas em residuais (Onyutha, 2020), o que é feito neste trabalho.

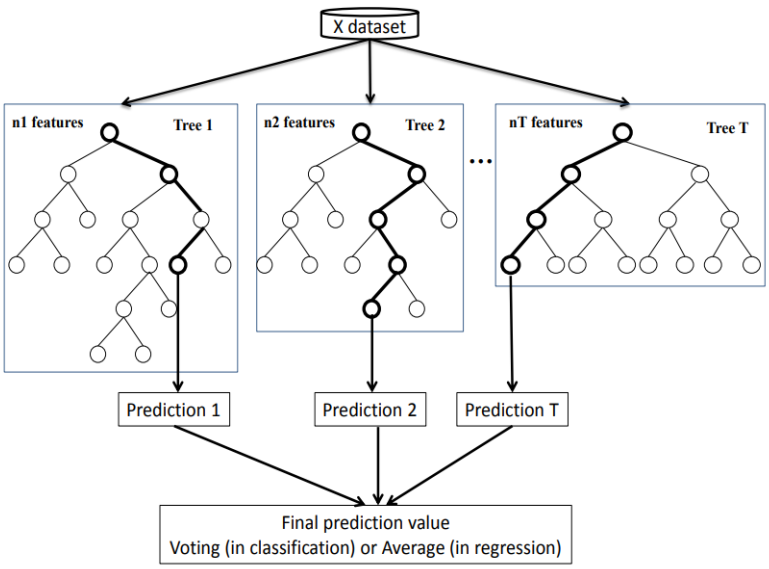
2.2 Métodos selecionados

São três os métodos de ML utilizados neste trabalho, escolhidos pela sua boa aplicabilidade em problemas de regressão sobre séries temporais.

2.2.1 *Random Forest*

O algoritmo de florestas aleatórias (do inglês *Random Forest*, ou simplesmente RF) é um método do tipo *ensemble*, que são aqueles que combinam os resultados de vários modelos em um só. Durante sua execução, são montadas diversas estruturas denominadas árvores de decisão, grafos acíclicos que representam uma sequência de decisões de forma hierárquica. Os nós desses grafos representam testes nos atributos dos dados de entrada, os ramos representam os resultados desses testes, e as folhas representam um resultado final, que será um rótulo em caso de problemas de classificação ou um valor numérico em

Figura 2.3 – Diagrama do método RF



Fonte: Nguyen et al. (2020)

caso de regressão (Liaw; Wiener et al., 2002).

O RF monta diversas árvores de decisão, sendo cada uma treinada sobre uma amostragem diferente do conjunto de entrada original, e ainda, para cada divisão de nó, são considerados apenas um subconjunto aleatório de atributos. A introdução desses comportamentos aleatórios, proposta por Breiman (2001), garante a diversidade entre as árvores e, se aplicada na medida correta, traz resultados acurados. As proporções de cada uma dessas divisões estão entre os hiperparâmetros do modelo.

Após o processo de construção, previsões são feitas pelo conjunto de árvores montadas, dependendo do tipo de problema que está sendo enfrentado. Quando se trata de classificação, a decisão é tomada por votação majoritária entre as árvores. Já quando se fala em problemas de regressão, é feita uma média entre as previsões individuais de cada árvore. A imagem 2.3 contém um diagrama do processo.

A regressão por RF não é comumente utilizada em problemas de série temporal, tendo em vista que o método envolve a divisão do *dataset* em subconjuntos independentes, quebrando a ordem dos dados e assim dificultando a observação das dependências temporais e padrões sequenciais subjacentes. Existem estratégias para contornar essa limitação, como transformações nos dados de entrada de forma a simular um problema de regressão comum, no entanto isso nem sempre é necessário.

Modelos baseados em árvores de decisão receberam significativa melhoria quando

combinados com o uso de *gradient boosting*, um método *ensemble* em que diversos preditores simples são treinados em sequência, aprendendo com os erros dos anteriores. Uma versão inicial do método foi proposta em 1990 e, já em meados daquela década, árvores de pouca profundidade mostraram-se uma boa escolha para o preditor simples exigido pela estrutura (Friedman; Hastie; Tibshirani, 2000).

A ideia por trás do *gradient boosting* é montar um modelo robusto composto por um conjunto de modelos mais fracos. Quando aplicado a árvores de regressão, o método vai adicionando iterativamente novas árvores de decisão ao modelo, ajustadas de acordo com os erros das predições do acúmulo de árvores existentes até então. Para a predição final, usa-se um cálculo sobre as predições de todo o conjunto de árvores (Elith; Leathwick; Hastie, 2008).

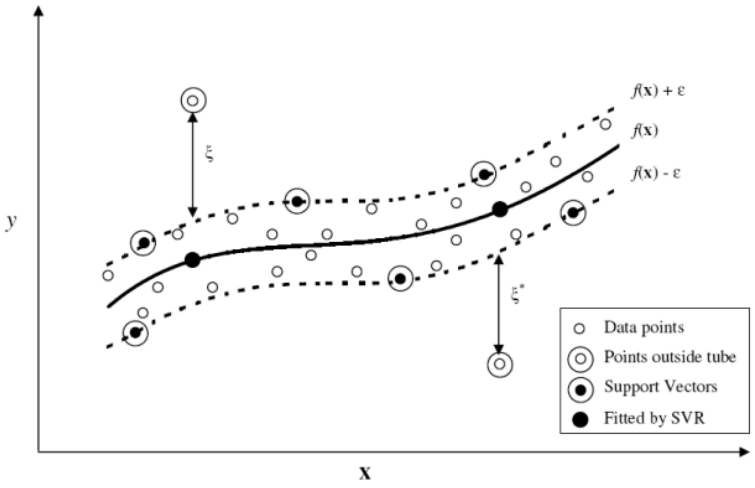
É a abordagem sequencial do *gradient boosting* que permite ao modelo final captar padrões complexos nos dados, como aqueles presentes em séries temporais. Como cada nova árvore é treinada sobre os resultados das anteriores, capturam-se padrões sazonais complexos nos dados. O poderio de *gradient boosted trees* (árvores construídas com *boosting*) tornou-as populares em competições de modelagem de séries temporais em anos recentes, e hoje há boas opções de bibliotecas que implementam-nas de forma acessível. Sendo assim, escolheu-se para este trabalho uma implementação de regressão por RF que inclui preceitos desse método.

2.2.2 *Support Vector Regressor*

O algoritmo *Support Vector Machine* (SVM) é um classificador que trata as instâncias do conjunto de treino como pontos em um espaço multidimensional e busca traçar o melhor hiperplano que as separe corretamente. Um hiperplano é uma superfície de decisão, sendo uma linha em caso de um espaço bidimensional ou um plano em mais dimensões, e a ideia é posicioná-lo de modo a separar as instâncias por categoria (Noble, 2006). Denomina-se margem o espaçamento entre o hiperplano e as instâncias mais próximas, essas denominadas vetores de suporte (do inglês, *support vectors*). O modelo buscará o hiperplano com a maior margem possível, que ao mesmo tempo mantenha o máximo de instâncias classificadas corretamente.

Support Vector Regressor (SVR) é uma variante de SVM aplicada em problemas de regressão, em que se busca definir não mais um hiperplano mas uma função contínua que melhor mapeie as instâncias de treino. Aqui a margem é entendida como um cilindro

Figura 2.4 – Diagrama do método SVR



Fonte: Lahiri and Ghanta (2008)

ou tubo que encapsula a função, e seu raio é definido pelo hiperparâmetro ε. Os vetores de suporte serão os pontos no limiar desse espaço, como mostrado na Figura 2.4. O modelo tentará definir a função mais plana possível que contenha em sua margem o máximo de instâncias (Awad et al., 2015).

O mapeamento de instâncias no espaço torna-se cada vez mais complexo e computacionalmente custoso quanto maior for a quantidade de atributos - com N atributos, serão N dimensões. Para contornar isso, SVRs utilizam uma função *kernel* que realiza operações sobre os dados de entrada sem convertê-los explicitamente em um espaço multidimensional (Gunn et al., 1998). A escolha do *kernel* é importante para os bons resultados do modelo, pois define o comportamento da função que será traçada.

O *kernel* mais simples é o linear, cuja equação está em (2.5). Existe também o *kernel* polinomial, que introduz não-linearidade ao elevar o produto dos vetores a um certo grau. Em sua equação (2.6), c é uma constante pré-definida e d é o grau do polinômio. Em ambas as equações, x_i e x_j são vetores de atributos de duas instâncias.

$$K(x_i, x_j) = x_i^T x_j \tag{2.5}$$

$$K(x_i, x_j) = (x_i^T x_j + c)^d \tag{2.6}$$

SVMs possuem forte sustentação teórica e trazem bons resultados em diversas aplicações (Noble, 2006). O SVR, embora não tão comumente utilizado quanto o classificador, possui estrutura robusta capaz de capturar as complexas relações não-lineares

entre atributos presentes em séries temporais, tudo de forma pouco custosa. Escolheu-se esse algoritmo por acreditar em sua capacidade de bem interpretar os dados coletados.

2.2.3 *Long Short-Term Memory*

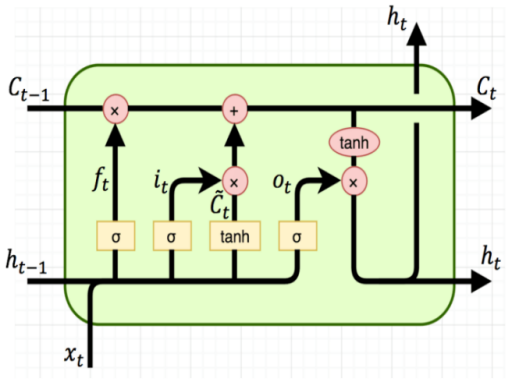
A arquitetura LSTM (do inglês, *Long Short-Term Memory*) pertence à família das redes neurais recorrentes (RNRs), cabendo aqui uma breve explicação de sua taxonomia. Redes neurais são compostas por nodos (ou nêuronios) agrupados em camadas, conectados entre si por "sinapses", em uma simulação das redes neurais biológicas. Durante o treinamento desses modelos, os dados são propagados desde a camada de entrada até a de saída um certo número de vezes (ou épocas), gerando previsões cujos erros são retropropagados na estrutura, para ajustes no ciclo seguinte.

As RNRs são um tipo específico de rede neural projetado para o uso em dados sequenciais, contendo, em suas camadas intermediárias (também chamadas de camadas ocultas), blocos de memória que retêm informações de leituras anteriores. O objetivo é prever o próximo passo em uma sequência com base nas últimas leituras feitas. No entanto, as RNRs tradicionais possuem uma memória de curto prazo, capaz de lembrar apenas alguns poucos passos anteriores (Siami-Namini; Tavakoli; Namin, 2018).

Essa limitação da memória de redes neurais tradicionais é responsável pelo problema da dissipação do gradiente, fenômeno que faz com que a RNR perca as informações que utiliza para se atualizar durante a etapa de treinamento. Redes neurais avaliam sua performance nessa etapa através da observação de uma função de perda, cálculo que quantifica diferenças entre valores reais e aqueles preditos pela rede. A cada época é obtido o gradiente da função de perda, vetor que dá informações de direção e ritmo dela e que é retropropagado na rede neural. O problema surge quando a sequência de dados é longa e o gradiente acaba por aproximar-se de zero e dissipar-se na rede, impedindo o aprendizado de dependências de longo prazo (Pascanu; Mikolov; Bengio, 2013).

Foi com o objetivo de solucionar esse problema que Hochreiter and Schmidhuber (1997) propuseram a arquitetura LSTM, composta por células de memória complexas, aptas a captar dependências de longo prazo nos dados. Assim como nas demais RNRs, essas células guardam informações do estado da leitura da sequência, mas diferenciam-se por possuir também um componente que guarda informações do estado da célula (Fathi, 2019). Esses dois componentes trabalham em conjunto, servindo como memórias de curto e longo prazo, respectivamente.

Figura 2.5 – Arquitetura interna de uma célula de LSTM.



Fonte: Nguyen et al. (2020)

A estrutura interna das células de memória do LSTM inclui ainda três *gates* que controlam o fluxo de informação através da rede. Há um *forget gate* que determina as informações que cada célula receberá da anterior, um *input gate* que determina o que será atualizado e um *output gate* que determina o que será passado adiante (Nguyen et al., 2020). A cada passo do processo de treinamento, o conjunto de dados passará por esse ciclo, formando espécie de filtro daquilo que o modelo considera relevante para o seu aprendizado. A Figura 2.5 exemplifica o esquema. Nela, h e C representam fluxos de informações sobre os dados e a célula em si, respectivamente; já f, i e o representam os *gates*.

Esses mecanismos de retenção de memória tornaram o LSTM escolha fácil para este trabalho. Ademais, trata-se de arquitetura altamente customizável, podendo ser adaptada aos mais diversos *datasets*.

3 TRABALHOS RELACIONADOS

Abaixo é feita uma revisão das fontes que inspiraram este trabalho, separadas por assunto. A Seção 3.1 foca em trabalhos de escopo geral, e a 3.2 em trabalhos que atacaram o mesmo problema desta pesquisa.

3.1 Uso de ML em predições de série temporal

Tradicionalmente, predições sobre dados de séries temporais são feitas com métodos puramente estatísticos. Algoritmos de ML passam a tornar-se mais atraentes para essa aplicação a partir da década de 80, quando é introduzida na área a técnica de retropropagação, que tornou o treinamento de modelos complexos computacionalmente tratável e embasa RNRs (Makridakis; Spiliotis; Assimakopoulos, 2018).

Desde então, houve considerável número de estudos analisando o poderio desses métodos, e comparando-os aos estatísticos tradicionais. Revisão bibliográfica feita por Ahmed et al. (2010) sobre estudos publicados desde 1995 aponta, porém, que os comparativos de mais larga escala feitos até então focavam quase exclusivamente em problemas de classificação, e que os resultados agregados eram pouco conclusivos.

Ao longo dos últimos anos, os rápidos avanços tecnológicos em *hardware*, com consequentes melhorias em poderio computacional e capacidade de armazenamento, renovaram o interesse no uso de ML em previsões, incluindo sobre séries temporais (Koparanov; Georgiev; Shterev, 2020). Nesse cenário, instauram-se competições de modelagem preditiva, dentre as quais destacam-se as edições de número 4 e 5 da *M-Competition*, especialmente relevantes para o trabalho atual. Trata-se de uma série de competições de construção de modelos para séries temporais, projeto idealizado por Spyros Makridakis, professor na Universidade de Nicósia no Chipre e pesquisador influente na área.

Após três edições entre 1982 e 2000 o evento passou por longo hiato, mas foi tomada a decisão de retomá-lo frente ao avanço da inteligência artificial ao longo do início do século XXI. Em artigo onde justificam o retorno do evento, os organizadores dizem que acreditam no potencial preditivo de ML, mas que há uma insuficiência dos métodos avaliativos utilizados até então, por isso a importância da realização de uma nova competição (Makridakis; Spiliotis; Assimakopoulos, 2018).

A quarta edição da competição foi realizada em 2018 e consistiu em avaliar os modelos submetidos pelos participantes sobre 100.000 séries temporais diferentes, que

variavam em frequência (desde diária até anual). Analisando os resultados, Makridakis, Spiliotis and Assimakopoulos (2020) identificam superioridade de modelos híbridos, que combinam atributos de ML com estatística, estrutura utilizada por nove dentre os dez primeiros colocados. Foi esta a primeira edição em que ML teve destaque, embora modelos de ML puros tenham obtido baixa performance comparados aos híbridos.

A edição seguinte, de 2020, baseou-se em *dataset* contendo vendas unitárias de 3049 produtos, ao longo de período de 5 anos, pela rede de verejo *Walmart*. Isso se deu em resposta a críticas recebidas pela edição anterior, apontando que o *dataset* que havia sido utilizado era demasiado esparso (Makridakis; Spiliotis; Assimakopoulos, 2022b). Dessa vez, houve predominância de métodos de ML entre os melhores colocados, muitos deles utilizando metodologias baseadas em *boosted trees*, conforme apontado em artigo de conclusão do evento (Makridakis; Spiliotis; Assimakopoulos, 2022a).

O que fica constatado pela análise dos resultados desses eventos é uma gradual melhoria em modelos de ML preditivos com foco em séries temporais, em anos recentes. As análises detalhadas publicadas pelos organizadores também apontam que os modelos respondem bem ao aumento do grau de concentração dos dados de treinamento. Tendo isso em vista, observa-se o quanto o contexto da epidemiologia no Brasil, em que se dispõe do robusto SINAN, é fertil para pesquisas do tipo.

3.2 Uso de ML em monitoramento de dengue

Assim como acontece com as previsões sobre séries temporais em geral de que se tratou na Seção anterior, o monitoramento de dengue tem histórico de uso de métodos puramente estatísticos com sucesso, mas ML vem despontando como boa ferramenta alternativa. Fazendo revisão histórica, Lourenço et al. (2018) apontam que modelos computacionais são aplicados em dengue desde meados do século XX, em tarefas gerais de monitoramento. Por tratar-se de vírus de grande diversidade genética e consequente *overlap* de sintomas com outras infecções, a construção de modelos encontra como obstáculo um certo grau de inconsistências nos relatos. De qualquer forma, os autores acreditam que sistemas robustos de vigilância possam suprir tais deficiências.

Uma revisão bibliográfica mais detalhada foi feita por Hoyos, Aguilar and Toro (2021), focando o período de 2015 a 2021. Dentro desse intervalo, os autores encontraram 64 artigos sobre uso de ML para o estudo de dengue. Os trabalhos cobrem diferentes aplicações e, dentre aqueles focados em modelagem de propagação da doença, RF e SVM

aparecem entre os três modelos mais utilizados. Em geral, os autores observam um padrão de relatos de bons desempenhos, mas com grande deficiência no detalhamento das técnicas de pré-processamento aplicadas. Sete dos artigos apurados são brasileiros.

Na listagem de artigos internacionais que atacaram o problema, Majeed et al. (2023) obtiveram bons resultados em previsões de casos de dengue a nível estadual na Malásia, após treinar uma variação de LSTM sobre conjunto encobrindo período de cinco anos. A mesma arquitetura foi usada por Nguyen et al. (2022) em previsões baseadas em dados climáticos do Vietnã, outro país onde a dengue é endêmica. Zhao et al. (2020) montaram regressor por RF para previsões a nível estadual na Colômbia, treinado com subconjuntos de dados nacionais, com bons resultados para até 12 semanas.

Dentre os trabalhos brasileiros que mais se aproximam do atual está o de Marques-Toledo et al. (2017), que consistiu na montagem de um modelo de predição de casos de dengue, treinado com dados do SINAN e com postagens sobre a doença na rede social Twitter. Os resultados relatados são bons, com capacidade de predição acurada para até oito semanas a nível de cidade. É interessante a escolha do uso de postagens, que de acordo com os resultados relatados mostraram-se dados úteis a esse tipo de análise, muito embora existam em número significativo somente em locais de maior desenvolvimento socioeconômico.

Já Stolerman, Maia and Kutz (2019) utilizaram dados climáticos de algumas capitais do país para o treino de seu SVM, em problema de classificação em que se buscava prever se determinado ano seria pandêmico naquelas cidades. A acurácia do modelo alternou no geral entre 60% e 80%, à exceção de uma das capitais, que apresentou 100%. Trata-se de bom precedente para o uso desse tipo de *dataset* e modelo no Brasil, mesmo que em aplicação e escopo diferentes do que o presente trabalho propõe.

Bastante relevante para a definição do escopo deste trabalho é o artigo de Mussumeci and Coelho (2020) que faz um comparativo entre os modelos LSTM, *LASSO Regressor* e RF, mais uma vez com dados climáticos a nível de cidade. O principal diferencial aqui, em relação aos estudos citados anteriormente, foi o escopo: 790 cidades em 5 estados. Os autores reconhecem que tratar dados cidade a cidade deixa de levar em conta o movimento populacional, que impacta diretamente no espalhar do vírus. Isso foi tratado na etapa de pré-processamento, construindo *datasets* sobre conjuntos de municípios vizinhos.

Por último, é necessário citar o artigo de Codeço et al. (2016) sobre a construção do sistema Infodengue, que automatiza o monitoramento da doença para centenas de

cidades no país. Através de um cálculo sobre *dataset* contendo informações de temperatura e números de postagens do Twitter, o sistema prevê o nível de alerta para uma dada semana, tratando-se aqui de um problema de classificação. Algo semelhante havia sido feito por pesquisadores da UFMG que construiram o sistema de monitoramento Observatório da Dengue, com auxílio de análise de sentimentos de postagens sobre dengue nessa mesma rede social (Silva et al., 2011). Considerou-se o uso do Twitter para o presente trabalho, tendo em vista a sua frequência na bibliografia, no entanto mudanças na API da plataforma dificultaram o acesso a dados, forçando o descarte desta opção [1].

Nota-se que há uma boa base de trabalhos em aplicabilidade de ML na área aqui estudada. Há outras vias de se atacar o problema, como o monitoramento da população de mosquitos, mas aqui foi mantido o foco em trabalhos que acompanham a manifestação da doença em si, como propõe o presente trabalho. A bibliografia dessa linha é recente e são muitos os rumos ainda não explorados. Ademais, o histórico de competições de ML estudado aponta para uma melhoria em performance dos métodos de ML existentes sobre séries temporais, o que inspira otimismo quanto aos resultados. No Brasil, os trabalhos que atacaram o problema focam majoritariamente em *datasets* recolhidos por município, por isso escolheu-se experimentar com a concentração por estado. Monta-se um método de *ensemble* por macrorregião como forma de explorar mais a fundo a possibilidade de previsões nessa escala. Como alternativa a dados de postagens no Twitter, adicionam-se dados do Google Trends ao *dataset*. Ainda, há deficiência em descrição de métodos de pré-processamento nos trabalhos revisados, etapa que se buscará descrever em nível adequado de detalhes.

[1] Planos atuais limitam severamente tarefas de *scraping*, como exposto na documentação da API em <https://developer.twitter.com/en/docs/twitter-api/getting-started/about-twitter-api>

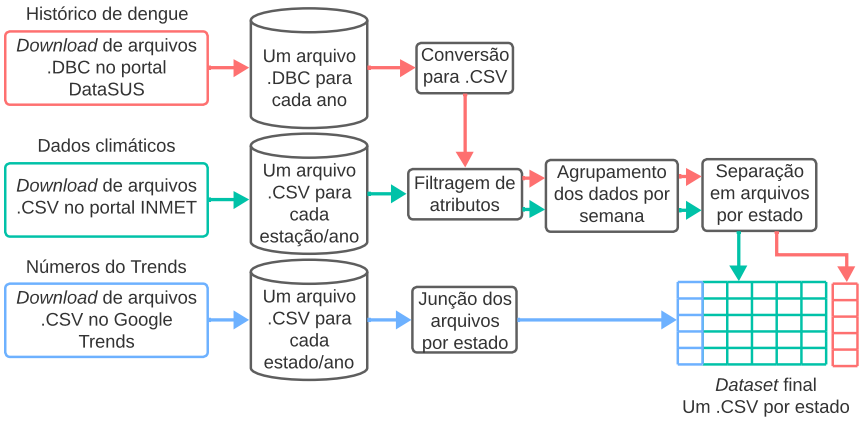
4 METODOLOGIA

Este trabalho é composto pela construção e avaliação de modelos de ML como preditores de casos de dengue no Brasil a nível de estado. O conjunto de dados final foi montado com dados recolhidos de diferentes fontes, organizados em 27 arquivos - um por unidade federativa. Para cada um deles, treinaram-se um modelo de cada algoritmo selecionado. Após isso, testes foram realizados em duas etapas: predições de modelos individuais por estado, e predições em modalidade *ensemble* com modelos agrupados por macrorregião. Neste capítulo busca-se detalhar o processo o melhor possível.

4.1 Coleta e pré-processamento dos dados

O *dataset* utilizado consiste em dados climáticos e do Google Trends como variáveis independentes, e o número de casos de dengue como atributo alvo. Optou-se por concentrar o trabalho sobre o período entre 2010 e 2022, a fim de montar *dataset* que fosse ao mesmo tempo recente e de tamanho considerável para o bom treinamento dos modelos. Ademais, dados climáticos anteriores ao período citado possuem consideráveis lacunas. O fluxograma em 4.1 mostra o procedimento de coleta e pré-processamento.

Figura 4.1 – Fluxograma da montagem dos *datasets*



Fonte: o autor

Na área de epidemiologia, a fim de estabelecer uma padronização para facilitar tarefas de vigilância, convencionou-se, internacionalmente, agrupar os dados por unidades de tempo chamadas semanas epidemiológicas. Trata-se de um período de sete dias,

Figura 4.2 – Visualização da versão final dos *datasets*

semana	temp. máx.	temp. mín.	temp. média	precipitação	umidade mín.	umidade média	vento máx.	vento média	trends	casos
201001	38.3	7.9	23.85	271.6	19	74.21	21.5	3.1	16	10
201002	40.8	5.7	23.57	286.31	7	74.47	28.8	2.94	16	11
201003	41.5	11.2	23.88	338.45	7	74.47	21.9	2.85	21	20
201004	41.2	6.8	23.67	347.97	12	73.57	22.6	2.86	0	13
● ● ●										
202251	37.3	-2.3	18.06	423.71	10	76.46	41.6	3.03	5	1
202252	38.3	0.4	17.87	315.37	16	76.94	35.5	2.98	8	2

iniciado no domingo, podendo ser 52 ou 53 ao longo do ano. Tendo em vista isso, além de peculiaridades da interface do Google Trends que dificultam a retirada acurada de dados a nível diário para períodos longos, agrupou-se o *dataset* por esta unidade de tempo. O período em análise, entre 03/01/2010 e 31/12/2022, totaliza 678 semanas epidemiológicas [1].

A Figura 4.2 mostra o estado final de um dos *datasets* gerados ao fim desse processo. Foram montados 27 arquivos como o da imagem, um para cada unidade federativa, contendo as supramencionadas 678 semanas[2], cada uma com 9 variáveis independentes. A coluna "semana"contém o índice para cada semana epidemiológica, desde a primeira de 2010 até a 52ª (última) de 2022.

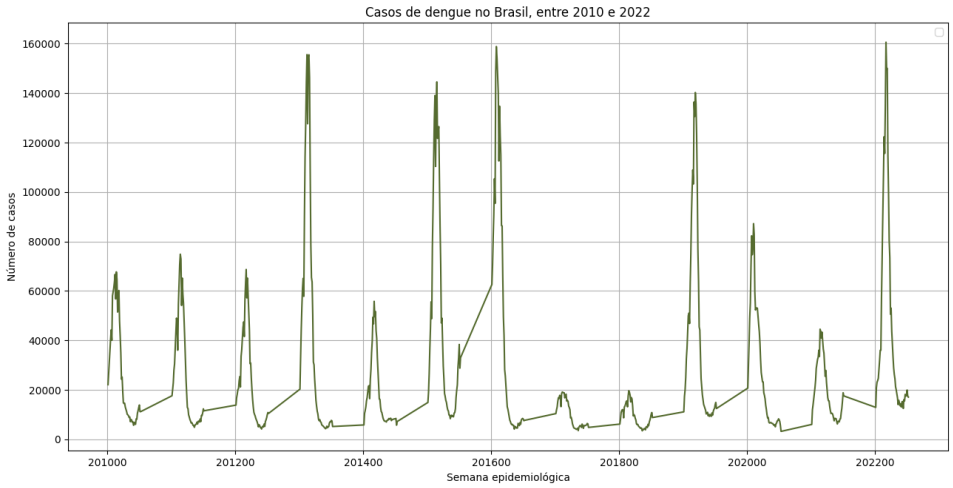
As manipulações que resultaram no conjunto final foram feitas em Python, com a biblioteca Pandas. A seguir explicam-se o processo e as peculiaridades para cada tipo de dado coletado.

4.1.1 Histórico de dengue no Brasil

O histórico de casos de dengue no Brasil, recolhido pelo Sistema de Informação de Agravos de Notificação (SINAN), é disponibilizado através do DATASUS[3], portal atrelado ao Ministério da Saúde. O portal disponibiliza um arquivo para cada ano, em formato DBC. Após o *download* manual dos arquivos, eles foram convertidos para o

[1] Calendários epidemiológicos em <http://portalsinan.saude.gov.br/calendario-epidemiologico>
[2] À exceção do *dataset* do estado de Roraima que continha apenas 644, o que é explicado em 4.1.2.
[3] Página de *downloads* em <https://datasus.saude.gov.br/transferencia-de-arquivos/>

Figura 4.3 – Casos de dengue no Brasil durante o período em estudo



Fonte: SINAN.

formato CSV com uso do pacote read.dbc[4] da linguagem R. Para cada caso relatado, os arquivos originais contêm atributos com informações de sintomas e localidade de origem. Foram mantidos apenas os números de casos, separados por estado. Os detalhes clínicos foram descartados por fugir do escopo do trabalho.

Por tratar-se de doença de notificação compulsória, são incluídos no sistema não apenas os casos confirmados mas também aqueles de suspeita, posteriormente atualizados caso exames confirmem tratar-se de outra enfermidade. Esse controle é mantido através de um atributo "evolução do caso". Considerou-se manter no *dataset* apenas casos anotados como confirmados, mas inconsistências no atributo, nem sempre preenchido, fizeram com que se optasse por manter todas as entradas. Ao todo, foram 19.973.144 casos relatados em todo o Brasil no período, com histórico traçado na Figura 4.3. O traçar desse histórico permite-nos observar que os casos distribuem estão distribuídos em períodos claros de altas, sempre coincidindo com o verão.

4.1.2 Dados climáticos

Os dados climáticos foram coletados do portal do Instituto Nacional de Meteorologia (INMET), onde são disponibilizadas leituras feitas por estações meteorológicas de

[4]Disponível em <https://github.com/danicat/read.dbc>

todo o país[5].

O portal disponibiliza arquivos individuais por estação meteorológica, por ano, em formato CSV, contendo leituras automáticas feitas de hora em hora. O código Python utilizado na manipulação agrupou-os para um arquivo por estação e alterou a frequência dos dados de horária para semanal, realizando os cálculos de médias, máximas e mínimas de acordo. Após isso, foram agregados os arquivos por estado, recalculando novamente esses números.

A seguir foi feita uma limpeza nesses dados. Utilizou-se o método VarianceThreshold da biblioteca Sklearn para verificar a existência de atributos com pouca ou nenhuma variação em seus valores e descartá-los, já que eles não contribuiriam para o aprendizado dos modelos. Também foi analisada a correlação de Pearson, métrica que avalia cada par de atributos, mensurando o quão fortemente correlacionados eles estão. Quando a métrica aproxima-se de 1 para dado par de atributos, significa que ambos movem-se na mesma direção e pode-se descartar um deles por ser redundante. É de ser reforçado que, para os objetivos aqui traçados, os *datasets* de cada estado deveriam ter todos os mesmos atributos. Ao final da limpeza, mantiveram-se os seguintes:

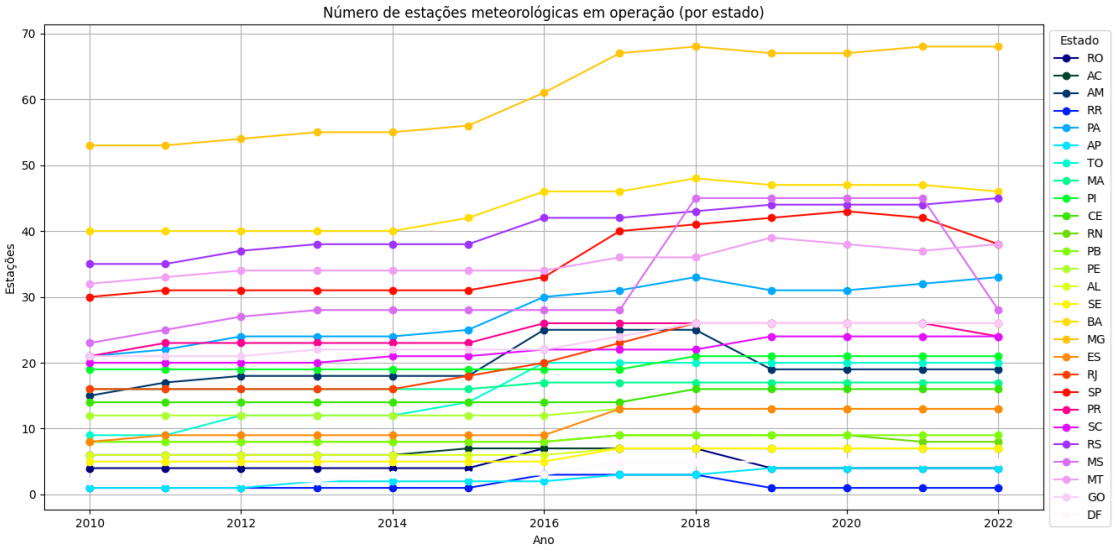
- Temperaturas: máxima, mínima e média, em graus Celsius;
- Precipitação: acúmulo, em milímetros;
- Umidade relativa do ar: mínima e média, em porcentagem;
- Rajadas de vento: máximas e médias, em metros/segundo.

Há grandes variações na qualidade das informações disponibilizadas pelo INMET, entre os diferentes estados. A começar pelo número de estações meteorológicas em atividade no período: Roraima possuía o menor número de estações em atividade, com apenas uma entre 2010 e 2015, e outras duas iniciando atividades em 2016. Em Minas Gerais, por outro lado, são listadas 69 estações. Uma menor quantidade de estações significa que os números obtidos não serão necessariamente representativos do clima do estado como um todo, o que pode causar impacto negativo sobre as previsões. A Figura 4.4 traça o histórico de estações em atividade no período, por unidade federativa.

Nessa linha, os dados de Roraima mostraram-se problemáticos: a leitura mais antiga disponibilizada para o estado está inserida na 35ª semana epidemiológica de 2010, que inicia-se no dia 29 de agosto daquele ano. O que significa que o *dataset* desse estado cobre apenas 644 semanas epidemiológicas, e não as 678 dos demais, sendo esse o único

[5] *Downloads* disponíveis em <https://portal.inmet.gov.br/dadoshistoricos>

Figura 4.4 – Histórico de estações meteorológicas ativas por estado



Fonte: o autor

conjunto a não encobrir toda a extensão delimitada na proposta.

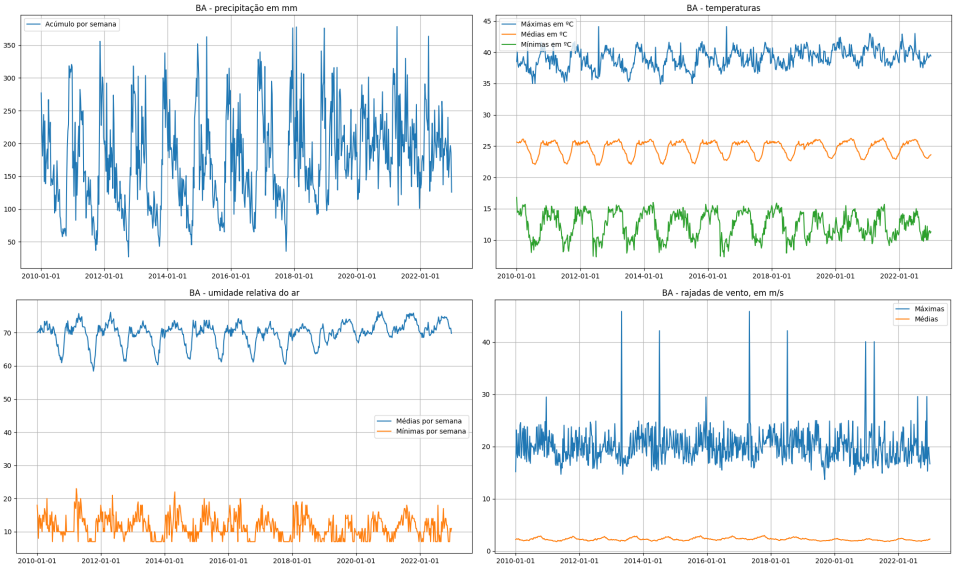
Ainda, ocasionalmente encontravam-se leituras inválidas nos arquivos, denotadas por linhas vazias ou preenchidas por um valor *default*, que foram desconsideradas. Isso não foi grande óbice à operação de montagem da maioria dos *datasets* finais, visto que a grande frequência das leituras, juntamente com a existência de múltiplas estações na maioria dos estados, garantiu que sempre houvesse ao menos uma leitura válida para cada semana epidemiológica. Ainda assim, trata-se de fator que contribui para o aumento da imprecisão dos dados e deve ser apontado. Feitas essas considerações, o *dataset* climático típico exibia padrões como o da Figura 4.5.

4.1.3 Números do Google Trends

Números de pesquisas contendo a palavra "dengue"no motor de buscas Google foram obtidos através do portal Google Trends. Esse sistema possibilita ao usuário pesquisar a popularidade de termos em uma cidade, estado ou país, dentro de um período específico, mas apresenta algumas limitações que exigem atenção especial. Ele não disponibiliza números concretos de buscas, mas sim uma escala relativa entre 0 e 100, onde 100 corresponde ao pico de pesquisas dentro do período selecionado [6]. A escala é cal-

[6]Documentação da plataforma em <https://support.google.com/trends/answer/4365533?>

Figura 4.5 – Exemplo do *dataset* climático típico. Dados recolhidos para o estado da Bahia.



Fonte: o autor.

culada com base em um subconjunto aleatório de buscas no território selecionado, de modo que ao fazer a exata mesma consulta mais de uma vez, o usuário obterá números diferentes.

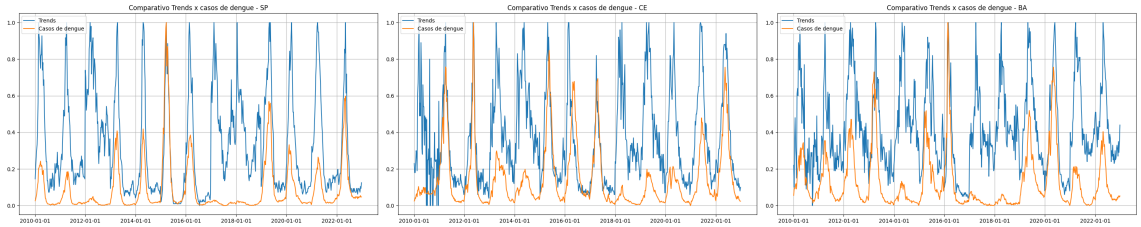
Ao realizar uma pesquisa no Google Trends, o usuário pode selecionar uma faixa de tempo qualquer para ser analisada, no entanto a frequência dos dados retornados dependerá da extensão dessa faixa. Para um período de até cinco anos, podem ser recolhidos números em frequência semanal. Acima disso, a agregação será por mês. Existem ferramentas externas que permitem contornar essa limitação, mas por questão de confiabilidade optou-se por utilizar a interface oficial. Para a montagem do *dataset*, foram feitas pesquisas individuais por ano epidemiológico, exportadas para CSV e depois concatenadas por estado.

Apesar das limitações, o Google Trends é uma fonte de pesquisa valiosa, em função do grande volume de informações, com dados globais mostrando que o Google mantém 90% do mercado de buscas na Internet desde 2010 [7]. Levantamento feito por Jun, Yoo and Choi (2018) encontrou 657 publicações que utilizaram a ferramenta em alguma forma, entre 2006 e 2016, em diversas áreas. Em epidemiologia, Ginsberg et al. (2009) montaram sistema de monitoramento do vírus Influenza sobre robusta base de milhões de pesquisas retiradas dela.

[7] Extraído de <https://gs.statcounter.com/search-engine-market-share#monthly-201001-202312> em 05/01/2024.

Ao contrastar os números dessa fonte com os casos de dengue relatados para cada estado, percebeu-se que há correlação entre o aumento de buscas no Google e períodos de alta da doença. Isso está exemplificado na Figura 4.6, em que, para cada exemplo, o índice do Google Trends está em azul e os casos de dengue em laranja. Para fins dessa comparação, foi necessário normalizar os números, uma vez que as escalas eram bastante diferentes.

Figura 4.6 – Comparativos entre índices do Google Trends e casos de dengue em diferentes estados, com os dados normalizados.



Fonte: o autor.

4.2 Montagem e treinamento dos modelos

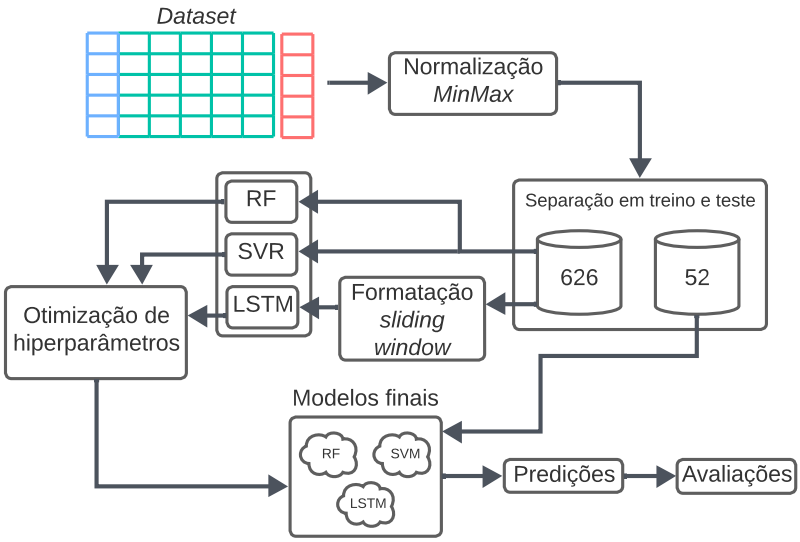
Todo o processo envolvendo os modelos foi feito em ambiente Python. Bibliotecas específicas serão mencionadas quando relevantes. Para cada um dos 27 *datasets* foram treinados três modelos, sendo um de cada algoritmo selecionado, em um total de 81. Antes do treinamento em si, foram efetuados alguns preparativos finais. Os procedimentos descrito nesta Seção estão representados no fluxograma em 4.7.

4.2.1 Preparativos para o treinamento

Os *datasets* foram normalizados com o método MinMaxScaler da biblioteca Sklearn, que redimensiona os atributos para o intervalo entre 0 e 1, assim garantindo que estejam todos na mesma escala. Decidiu-se por analisar a capacidade preditiva para o período de um ano, assim os dados foram divididos na proporção de 626 semanas para treino e 52 para teste.

O treinamento para LSTM exige uma transformação extra nos dados, pois faz uso de *sliding window*, o que exige que se moldem as variáveis independentes em um vetor de matrizes, em que cada índice contém n leituras, sendo n o tamanho da janela (em

Figura 4.7 – Fluxograma da montagem e treinamento dos modelos. Processo repetido para cada um dos *datasets*.



Fonte: o autor.

inglês, *lookback window*). A escolha desse tamanho tem grande impacto na performance: quando maior, facilita a captura de dependências de longo prazo, mas também torna o processo mais custoso. Para dados com padrões curtos e abruptos uma janela pequena basta, inclusive podendo acontecer uma queda repentina de acurácia com n acima de determinado valor (Kahraman et al., 2019).

Essa observação é corroborada por Koparanov, Georgiev and Shterev (2020), em experimento no qual um modelo foi testado sucessivamente sobre valores crescentes de *lookback window*, mantendo os demais fatores fixos. Os autores apontam a importância do equilíbrio entre a janela e a quantidade de épocas para o treinamento, havendo perda constante de acurácia acima de certa faixa de valores. Tais leituras auxiliaram na definição de uma faixa de busca para a *lookback window* ideal para este trabalho. Fixou-se a janela em 26 leituras após a execução de testes durante o processo de otimização, que será explicado na Seção seguinte.

4.2.2 Otimização de hiperparâmetros

A seleção dos melhores hiperparâmetros para os modelos foi feita com a biblioteca Optuna, proposta por Akiba et al. (2019), que implementa um método de otimização

bastante eficaz, pouco custoso em questão de recursos computacionais e altamente escalável. O método consiste em montar uma função objetivo, que recebe como entrada o conjunto de hiperparâmetros a ser otimizado e executa sucessivos testes, buscando minimizar ou maximizar determinada métrica de avaliação. Cabe ao usuário definir o espaço de hiperparâmetros a ser explorado e a métrica a ser otimizada. Para o trabalho, definiu-se como objetivo do otimizador a minimização do MSE, a fim de evitar erros gritantes nas previsões.

A função objetivo do Optuna realiza validação cruzada em seus testes, o que exigiu adaptações por se estar lidando com dados de série temporal. Utilizou-se o método TimeSeriesSplit da biblioteca Sklearn para efetuar a técnica como descrita na Seção 2.1.3, assim respeitando a sequencialidade dos dados.

A arquitetura LSTM é a mais complexa utilizada no trabalho, o que exigiu tratamento diferenciado nessa etapa, em comparação aos demais métodos. A exploração exaustiva do espaço de hiperparâmetros para todos os modelos a serem montados mostrou-se proibitivamente demorada, então optou-se por realizá-la inicialmente sobre um subconjunto dos *datasets* para, então, observada convergência entre essas estruturas, passar-se à montagem dos demais.

Para essa exploração inicial mais completa separaram-se cinco dos *datasets* que continham grandes diferenças na distribuição do atributo alvo, buscando encobrir a diverdidade do conjunto de dados: Amapá, Ceará, Minas Gerais, Pará e Santa Catarina. Ao fim dessa etapa fixaram-se a quantidade de camadas ocultas e respectivos nêuronios. Demais hiperparâmetros foram refinados para cada *dataset*.

4.2.3 Estrutura final dos modelos

A implementação dos regressores por RF foi feita com o método XGBRFRegressor da biblioteca XGBoost [8], que é bastante popular em ML . Trata-se de uma implementação proprietária de RF com base no método de *boosting* inicialmente proposto por Chen and Guestrin (2016), que também dá nome à biblioteca. Otimizaram-se os seguintes hiperparâmetros:

- *n_estimators*: número de árvores;
- *max_depth*: profundidade máxima de cada árvore;

[8] Biblioteca disponível em <https://github.com/dmlc/xgboost/>

- *subsample*: proporção de instâncias usadas no treino delas;
- *colsample_bytree*: fração de atributos considerados na construção de cada uma.

Os dois primeiros da lista lidam diretamente com a complexidade do modelo e os dois últimos afetam o grau de aleatoriedade que é introduzido durante a montagem. Cabe apontar que um modelo altamente complexo nem sempre é desejável: uma das possíveis causas para *overfitting* é o excesso de robustez do modelo em comparação à qualidade dos dados de treinamento.

Os modelos de SVR foram montados com o método de mesmo nome da biblioteca Sklearn. Para esses, otimizaram-se os seguintes:

- *kernel*: formato da função;
- C: grau de tolerância a erros;
- ε: largura da margem.

Os parâmetros C e ε trabalham juntos para equilibrar a complexidade e a acurácia do modelo. Com um C baixo e uma margem larga arrisca-se montar um modelo flexível demais, em que erros não serão suficientemente penalizados.

A tabela 4.1 resume as estruturas de todos os modelos de SVR e regressor por RF que foram montados. Incluem-se, para os hiperparâmetros numéricos, os valores mínimos e máximos registrados.

Tabela 4.1 – Hiperparâmetros para SVR e regressor por RF

Algoritmo	**Hiperparâmetro**	**Faixa de valores**
SVR	*kernel*	[*poly*, *linear*]
	ε	[0.00334, 0.00559]
	C	[0.02405, 0.04441]
RF	*n_estimators*	[100, 150]
	max_depth	[3, 8]
	subsample	[0.53523, 0.8989]
	colsample_bytree	[0.52586, 0.80964]

Para a montagem dos modelos de LSTM utilizou-se a biblioteca Keras[9]. Os hiperparâmtros otimizados foram:

- o número de camadas ocultas;
- *lookback window*: tamanho da *sliding window* aplicada aos dados de entrada;
- *units*: a quantidade de nêuronios por camada; é necessário otimizar um parâmetro desses para cada camada da rede;

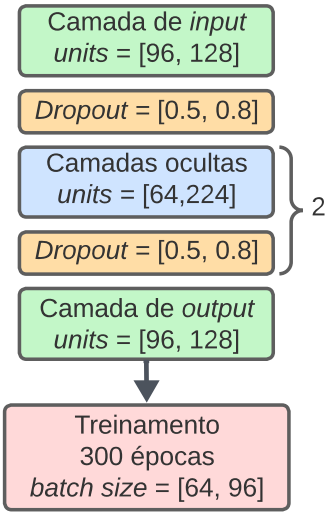
[9] <https://github.com/keras-team/keras>

- taxa de *dropout*: mecanismo para impedir *overfitting*;
- *batch size*: a quantidade de instâncias lidas pelo modelo a cada passo do treino.

Da lista acima, os dois primeiros parâmetros foram definidos sobre um subconjunto dos *datasets*, separados conforme descrito na Seção 4.2.2. A otimização do número de camadas ocultas foi o que demandou mais esforço computacional, e felizmente esse subconjunto inicial convergiu para uma mesma quantidade. O terceiro item da lista, a taxa de *dropout*, faz com que a cada passo do treinamento seja ignorada uma parcela dos nêuronios, de modo a fazer com que o modelo não se adapte demais a ruídos dos dados (Srivastava et al., 2014). Aplica-se esse mecanismo através da inserção de camadas de *dropout* entre as demais.

Estabaleceu-se um limite de 300 épocas para o treinamento, em conjunto com a aplicação de *early stopping*, prática que cessa o treinamento antecipadamente caso perceba-se que o modelo não está mais aprendendo com os dados. O monitoramento desse processo é feito pela observação de uma função objetivo, ocorrendo a interrupção, portanto, quando os valores dessa função estagnam ou deterioram. Utilizou-se como função objetivo o cálculo de MSE, prática comum na construção de modelos de regressão. A combinação de um valor relativamente alto de épocas com o uso de *early stopping* tem a finalidade de garantir que o treinamento efetivamente continue até que não haja mais progresso. A Figura 4.8 mostra a arquitetura final dos modelos de LSTM.

Figura 4.8 – Estrutura final dos modelos de LSTM



5 RESULTADOS

Apresentam-se neste capítulo os resultados dos experimentos, em números e em representações gráficas. Após uma visão geral das métricas, são traçados comparativos entre os métodos e separados alguns *datasets* de destaque para uma análise mais a fundo.

5.1 Previsões individuais por estado

Nesta Seção é feito um resumo dos resultados obtidos nas previsões individuais para cada estado. Inicialmente, as métricas recolhidas são listadas em tabelas, destacando as melhores dentre elas, e após isso é feito um comparativo entre os métodos utilizados. Tendo em vista o grande número de modelos que foram montados, tenta-se identificar padrões nos resultados e, com base neles, destacam-se predições que se mostraram particularmente interessantes.

A tabela 5.1 contém os valores de MAE que foram registrados, com os melhores para cada método em destaque. Nesta métrica todos os modelos apresentaram performances semelhantes estado a estado, e os três métodos obtiveram seus melhores números sobre os mesmos *datasets*. Os números chegam à casa dos milhares em certos casos, como em Goiás, Mato Grosso, Paraná e São Paulo. À primeira vista são observações preocupantes, mas para o problema em questão a captura de tendências pode ser mais importante que a previsão de números exatos. Ademais, a caracterização de um MAE como alto ou baixo depende da dimensão assumida pela variável dependente.

Há algumas exceções pontuais a esse equilíbrio, em que um dos métodos se destaca em relação aos outros. SVR obteve números relativamente altos para Minas Gerais e Paraná, por exemplo. Já o contrário ocorre com os números de RF para São Paulo e Mato Grosso do Sul, e os de LSTM para Ceará e Santa Catarina, significativamente mais baixos que os demais. O padrão geral, porém, não aponta em definitivo a superioridade de um método em relação aos demais, ao menos nessa métrica.

A tabela 5.2 exibe os valores de R^2 e MAPE obtidos. Essas métricas são exibidas em conjunto por serem ambas independentes da escala das variáveis, permitindo assim um melhor comparativo. Para cada algoritmo e métrica foram destacadas as cinco melhores performances, a fim de oferecer um panorama geral da capacidade de cada método.

Percebe-se que há uma inconsistência em performance dentro de cada método, com ambas as métricas exibindo valores entre o bom e o péssimo. Já uma análise estado

Tabela 5.1 – Valores de MAE para todos os modelos. Destacam-se os cinco melhores colocados, por método.

Estado	RF	SVR	LSTM
AC	180,571	150,089	111,914
AL	558,605	626,438	429,711
AM	**119,356**	**83,733**	**108,155**
AP	**5,849**	**7,594**	**8,284**
BA	416,038	496,245	514,218
CE	1262,294	1177,490	881,476
DF	1162,419	1384,093	1194,501
ES	402,974	272,718	554,488
GO	3122,785	3220,803	3350,433
MA	**127,577**	**108,071**	**100,044**
MG	1684,294	2402,042	1804,807
MS	388,753	548,077	665,738
MT	624,166	774,612	315,620
PA	**88,067**	**108,149**	**106,517**
PB	567,102	650,506	494,039
PE	678,963	610,635	511,221
PI	627,839	641,098	642,581
PR	2489,436	4412,389	3136,095
RJ	384,865	237,568	308,343
RN	736,542	708,776	775,121
RO	305,72	356,222	311,088
RR	**29,254**	**23,630**	**29,966**
RS	1843,881	1866,641	1694,017
SC	2531,350	2524,365	2179,177
SE	156,572	187,734	172,673
SP	4921,770	7070,754	7490,209
TO	536,004	635,986	374,544
Médias	961,223	1158,757	1046,851

a estado permite observar certa constância: para alguns deles foram obtidos resultados relativamente bons ao longo de múltiplos algoritmos e em ambas as métricas, como é o caso de São Paulo, Pará, Mato Grosso do Sul, Ceará, e Tocantins. Já para outros, más leituras foram registradas, como em Espírito Santo, Santa Catarina, Acre e Roraima. Isso indica que alguns dos *datasets* são mais adequados para predições do que outros, seja pela qualidade dos dados ou pela inconsistência dos seus padrões temporais.

Os estados que se destacaram pela difícil modelagem estão também entre aqueles onde havia menos estações meteorológicas em atividade no período, o que pode ter feito com que os dados climáticos recolhidos não fossem de fato representativos do panorama para todo o território. O oposto é verdade para aqueles com leituras melhores. Nisso evidencia-se que a diferença em disponibilidade de dados entre os estados, percebida

Tabela 5.2 – Valores de R² e MAPE para todos os modelos. Destacam-se os melhores colocados dentro de cada algoritmo, por métrica.

	RF		SVR		LSTM	
estado	R²	MAPE	R²	MAPE	R²	MAPE
AC	-6,066	1,145	-4,293	0,907	-4,974	0,714
AL	0,171	1,083	-0,217	0,591	**0,462**	1,245
AM	-1,950	0,763	-0,941	**0,384**	-2,606	0,435
AP	0,041	**0,39**	-0,572	**0,412**	0,026	0,486
BA	**0,677**	0,519	**0,574**	0,639	0,428	0,567
CE	-0,082	0,632	**0,097**	0,522	**0,539**	**0,407**
DF	-0,492	0,617	-0,953	0,786	-0,476	0,708
ES	-4,406	2,246	-0,145	1,596	-3,012	2,357
GO	-0,249	**0,433**	-0,122	0,568	-0,415	0,579
MA	-0,441	1,184	-0,135	0,622	0,031	0,678
MG	0,349	0,603	-0,457	0,511	**0,452**	0,582
MS	**0,637**	**0,377**	-0,052	**0,354**	0,061	0,567
MT	-0,206	0,489	-0,527	0,640	**0,702**	**0,323**
PA	**0,354**	**0,383**	-0,083	**0,372**	0,035	**0,321**
PB	-0,286	0,682	-0,604	0,686	-0,305	0,471
PE	0,123	1,187	-0,018	0,538	0,041	**0,364**
PI	-0,307	1,604	-0,438	0,928	-0,433	0,967
PR	**0,663**	0,586	-0,178	-0,178	**0,456**	0,901
RJ	-0,822	2,541	**0,122**	1,090	-0,259	1,595
RN	0,013	0,943	**0,025**	0,554	-0,295	0,514
RO	-2,260	0,596	-2,922	0,741	-1,795	0,740
RR	-1,594	3,780	-0,546	2,694	-1,386	2,690
RS	-0,317	0,816	-0,335	0,870	-0,141	0,664
SC	-0,465	0,825	-0,477	0,790	-0,124	0,706
SE	-0,337	0,539	-0,878	0,658	-0,501	0,645
SP	**0,592**	**0,303**	**0,239**	**0,510**	0,290	0,731
TO	-0,339	0,436	-0,701	0,569	0,196	**0,429**
Médias	-0,620	0,951	-0,538	0,747	-0,534	0,792

durante a coleta, de fato impactou os resultados finais.

Lacunas nos dados originais podem ter sido fatores também. Embora para cada estado sempre houvesse ao menos uma leitura climática por semana, não se descarta a possibilidade de ter havido períodos de dias em sequência em que uma ou outra estação não estivesse em pleno funcionamento. Para o caso de Roraima, o fato de possuir 34 semanas de leituras a menos em relação aos demais estados é quase certamente a responsável pelos resultados. É interessante observar, porém, que as suas avaliações ainda mostraram-se melhores que as do Acre. Outro fator podem ser anomalias como surtos da doença, algo com que os modelos possivelmente tenham dificuldade para lidar.

Para alguns dos *datasets* é observada inconsistência entre as métricas. Em Amazonas e Goiás, são relativamente baixos os MAPEs, mas há R² negativos, sugerindo predi-

ções acuradas mas sem a captura de padrões subjacentes nos dados. O contrário acontece com os modelos de RF e LSTM para Alagoas, em que R^2 relativamente alto é acompanhado de MAPE também alto.

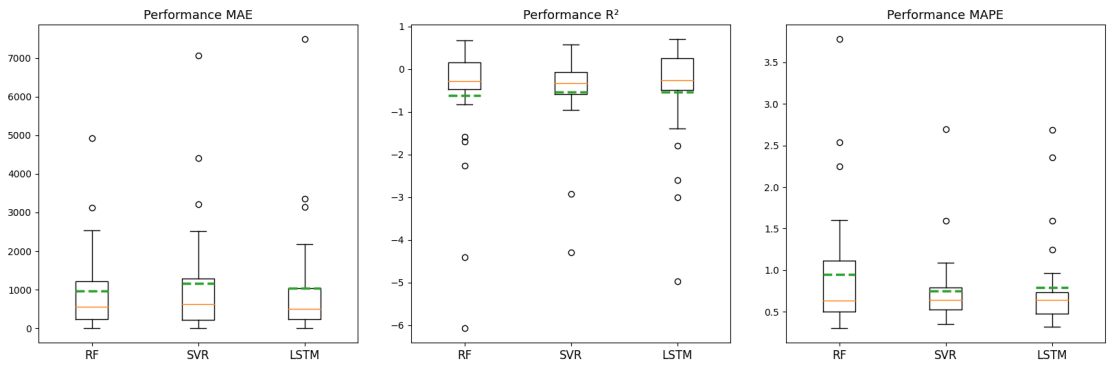
A maioria das leituras de R^2 obtidas foram negativas. É o caso de 17 resultados para RF, 22 para SVR e 14 para LSTM. Isso indica frequentes trechos em que os modelos falharam em capturar tendências: por exemplo, o número de casos sobe, mas o modelo prevê que desce, ou que não sobe com a mesma intensidade. Situações do tipo têm grande impacto nessa métrica e inevitavelmente ocorrerão quando a variável alvo for propensa a alterações repentinas, como é o caso no contexto aqui abordado.

O cenário desenhado por esse resumo permite um cauteloso otimismo. Em valores absolutos, os modelos performaram de forma equilibrada, com alguns poucos *outliers* chamando atenção. Porém, as demais métricas permitem crer que ao menos um subconjunto dos estados é mais promissor para modelagens do tipo. Os erros em porcentagem flutuam na faixa de 30% para cima e, embora poucos sejam os placares de R^2 positivos, isso não foge muito do esperado para um contexto de variáveis tão instáveis.

5.1.1 Comparativos entre os algoritmos

Os *boxplots* da Figura 5.1 resumem as métricas, possibilitando um melhor comparativo entre os algoritmos. As diferenças observadas não são gritantes, não havendo total superioridade de um em relação aos demais, mas é possível tecer alguns comentários.

Figura 5.1 – Boxplots para MAE, MAPE e R^2, por método. As linhas verdes tracejadas apresentam as médias; as linhas laranjas, as medianas.



As distribuições em LSTM são em geral melhores, mesmo que por pouca dife-

rença, com números de MAE e MAPE bem concentrados em faixas de baixo valor e uma maior quantidade de R^2 positivos, indicando que esse método foi o que melhor conseguiu interpretar as relações entre os atributos. Porém, chamam atenção *outliers* extremamente ruins em R^2 e MAE, indicando que mais esforços poderiam ter sido empregados para garantir modelos que se adaptassem às peculiaridades dos diferentes *datasets*. Essa superioridade não surpreende, tendo em vista as características dessa arquitetura, mais complexa e melhor adaptada a séries temporais.

Os modelos de SVR não ficaram muito atrás, nas três métricas. Inclusive, a menor quantidade de *outliers* e a menor complexidade dos modelos em comparação com LSTM são fortes argumentos a seu favor, indicando ser um método ao mesmo tempo consistente e de mais fácil montagem. No entanto, em problemas desse tipo é importante a captura de padrões, campo em que LSTM se sobressaiu e que assegura que mantenha a superioridade.

Para RF vê-se uma diferença mais clara, sendo esse o método que se mostrou menos adequado para o problema. Embora apresente mais R^2 positivos em comparação com SVR, também apresenta muitos deles profundamente negativos, e MAPEs bastante ruins. Ao lado dessas métricas, os MAEs baixos podem até mesmo indicar *overfitting* em alguns dos testes. Quando da escolha dos métodos, sabia-se que RF poderia não ser o melhor ajustado para o problema, mas acreditou-se que as especificidades da biblioteca escolhida poderiam contornar isso, o que por fim não se confirmou.

É importante apontar que, para cada métrica, os *outliers* foram os mesmos, reforçando a observação anterior de alguns *datasets* serem melhor adequados a previsões do que outros. Para R^2, há em comum os estados do Acre, Rondônia, Espírito Santo e Amazonas; para MAPE, Roraima, Espírito Santo e Rio de Janeiro; para MAE, São Paulo, Goiás e Paraná. De qualquer forma, o grau de dificuldade de cada método com esses *outliers* é critério válido para comparação, e nesse quesito há pouca diferença entre SVR e LSTM.

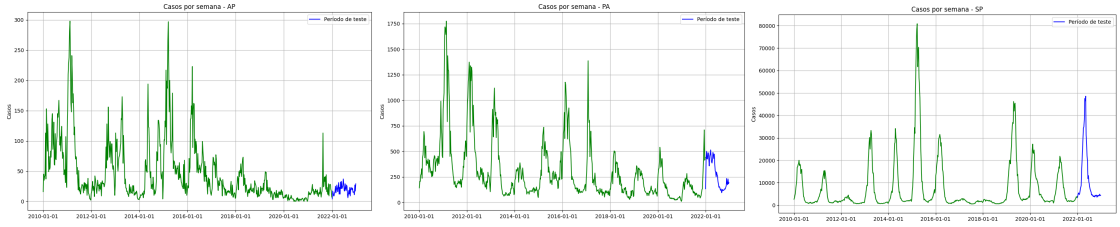
5.1.2 Estudos de caso - padrões gerais

Separam-se agora alguns dos estados que se destacaram nos resumos, seja pelo desempenho bom ou ruim, ao longo dos três métodos. Pela observação de gráficos, espera-se explicar o que possa ter causado os resultados.

A Figura 5.3 contém algumas das previsões que se destacaram por boa perfor-

mance em mais de um método, e a Figura 5.2 exibe a distribuição da variável alvo nesses *datasets*. Em Amapá e Pará os casos distribuem-se em um padrão serrilhado, com flutuações em curtos espaços de tempo que puderam ser bem capturadas. Para o Amapá o resultado disso foram MAE e MAPE particularmente baixos, e no Pará isso trouxe bons R^2. São Paulo apresenta períodos de alta bem definidos, padrão propício a predições. *Datasets* com ciclos bem claros, tanto na variável dependente quanto nas independentes, tendem a melhores resultados.

Figura 5.2 – Distribuição de casos para AP, PA e SP. Em azul, o período de 52 semanas separado para teste.



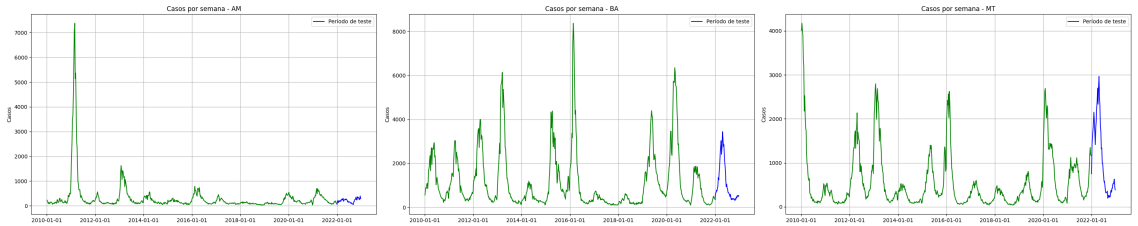
Fonte: o autor.

Figura 5.3 – Previsões para AP, PA e SP: exemplos de melhores resultados. Em azul, os números reais; em laranja, as predições.

A seguir, são destacadas algumas performances que foram boas em aspectos pontuais. Novamente, contrastam-se as distribuições em 5.4 com as previsões em 5.5. Para Amazonas, os valores de MAE e MAPE estão entre os mais baixos (à exceção do seu

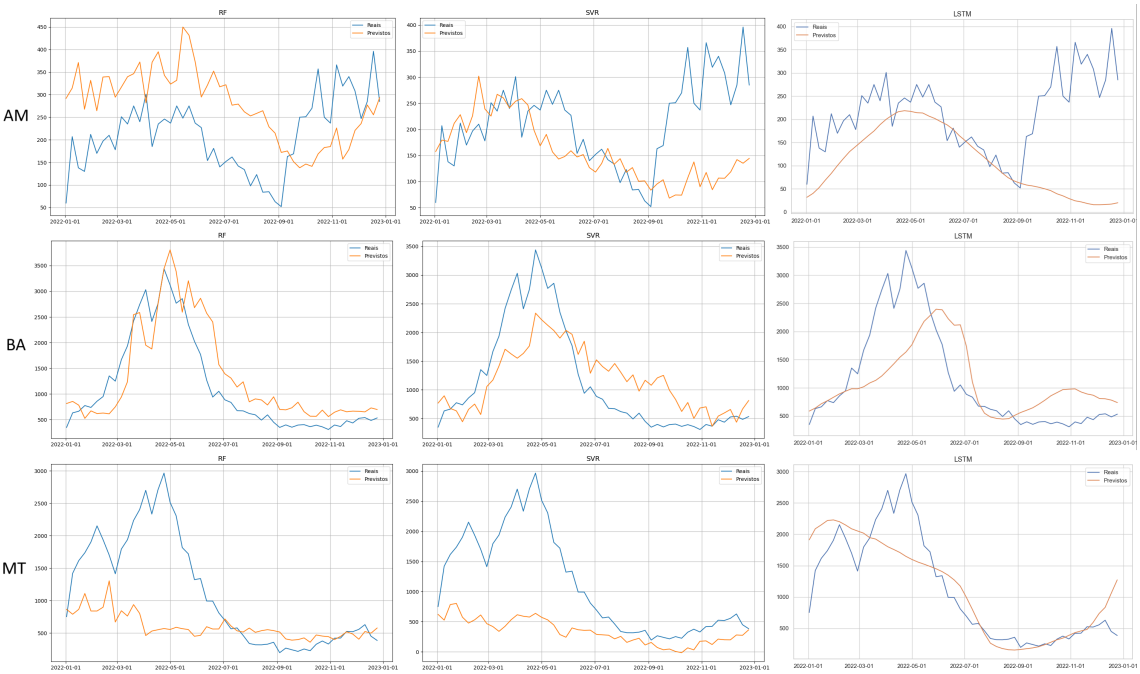
modelo RF), mas os R^2 estiveram entre os piores. O seu gráfico não denota padrão claro, o que pode explicar isso. A Bahia está destacada pelos consistentes R^2 positivos, na faixa entre 0.4 e 0.6, o que se explica pelo padrão cíclico bem definido em seu número de casos. Para o Mato Grosso obteve-se resultado especialmente bom com LSTM nas três métricas. Aqui também há um visível ciclo, mas com altos de dimensões variadas, o que foi melhor captado por essa arquitetura mais complexa.

Figura 5.4 – Distribuição de casos para AM, BA e MT. Em azul, o período de 52 semanas separado para teste.



Fonte: o autor.

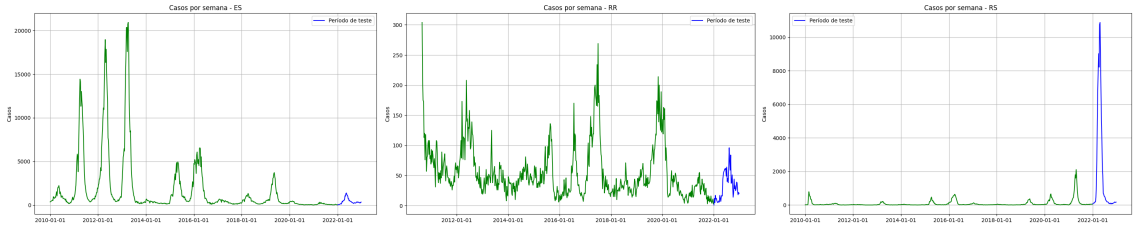
Figura 5.5 – Previsões para AM, BA e MT: exemplos de resultados bons. Em azul, os números reais; em laranja, as predições.



O próximo conjunto considerado de interesse contém casos em que nenhum método conseguiu atingir resultados satisfatórios. A análise das Figuras 5.6 e 5.7 mostra que nos três *datasets* o número de casos está distribuído de forma irregular. Em Espírito

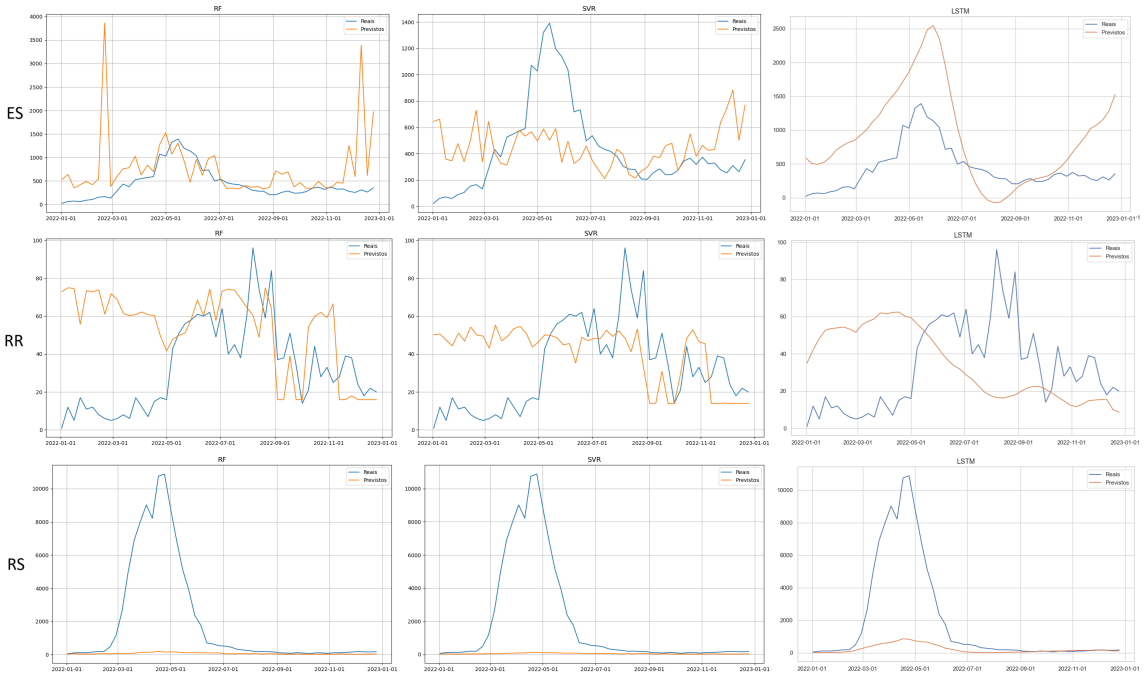
Santo o período de teste inclui números incomumente baixos, enquanto o contrário acontece com o Rio Grande do Sul. Já Roraima contém distribuição bastante ruidosa, o que se soma ao fato de possuir um menor conjunto de treino.

Figura 5.6 – Distribuição de casos para ES, RR e RS. Em azul, o período de 52 semanas separado para teste.



Fonte: o autor.

Figura 5.7 – Previsões para ES, RR e RS: exemplos de resultados ruins. Em azul, os números reais; em laranja, as predições.

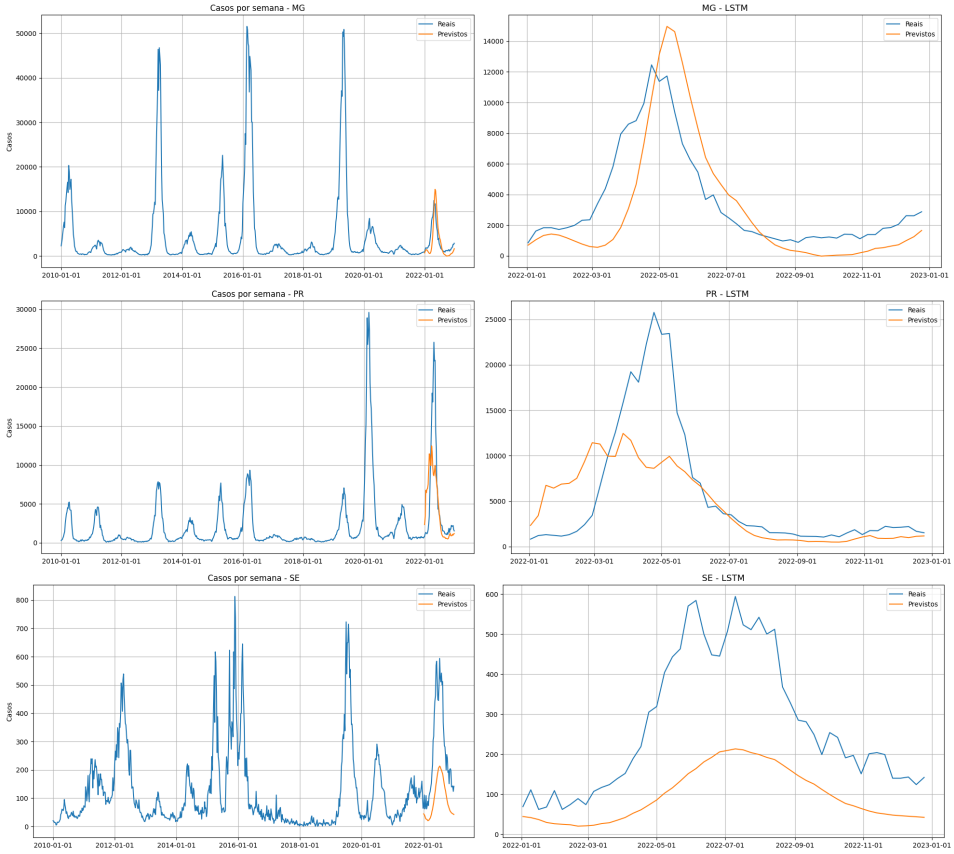


5.1.3 Estudos de caso - LSTM

Seguindo os esforços de resumir os padrões detectados nas previsões, nesta Seção são selecionados alguns dos resultados obtidos por LSTM, método que melhor se saiu nos experimentos. Para cada exemplo, as previsões são contrastadas ao *dataset* como um

todo, a fim de oferecer uma visão completa do cenário. A Figura 5.8 traz exemplos de

Figura 5.8 – Previsões por LSTM para MG, PR e SE. Na coluna da esquerda, as previsões contrapostas ao todo; na da direita, um *zoom* no período de teste.



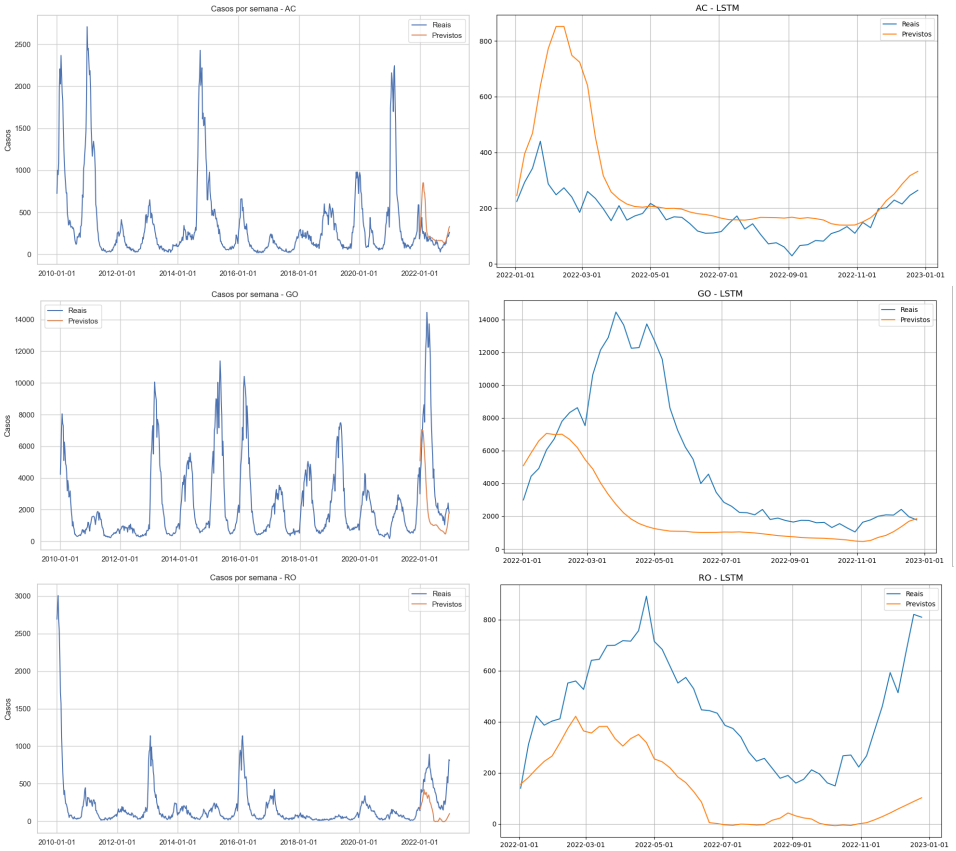
Fonte: o autor.

algo que foi observado com frequência: modelos que capturam tendências, mas erram quanto aos valores. Nos exemplos da imagem, é identificada a alta de casos, mas não a intensidade. De qualquer forma, Minas Gerais e Paraná apresentaram R^2 relativamente bons, o que indica a adequação dos modelos. Ambos estão entre os MAEs mais altos registrados, mas não preocupam no contexto desenhado: em Minas Gerais se está lidando com números altos em todo o período, não se traduzindo o MAE alto para um MAPE alto; e no Paraná, a alta de casos no período de teste foi anômala e portanto fugia do padrão esperado pelo modelo. Em Sergipe, onde o erro é mais intenso que nos outros dois listados, foi identificado R^2 negativo. Como dito anteriormente, captar tendências é, para este problema, tão ou mais importante quanto prever números, já que isso bastaria para ao menos alertar autoridades para possíveis surtos futuros. Essa tendência nos resultados é boa, embora ainda haja muito a melhorar.

Os gráficos em 5.9 repetem o comportamento de erro na intensidade dos altos e

baixos da variável alvo, dessa vez resultando em R^2 negativos para todos. Algo curioso é que para Goiás e Rondônia parece haver um adiantamento da curva pelos modelos, prevendo queda de casos antes dela efetivamente ocorrer.

Figura 5.9 – Previsões por LSTM para AC, GO e RO. Na coluna da esquerda, as previsões contrapostas ao todo; na da direita, um *zoom* no período de teste.



Fonte: o autor.

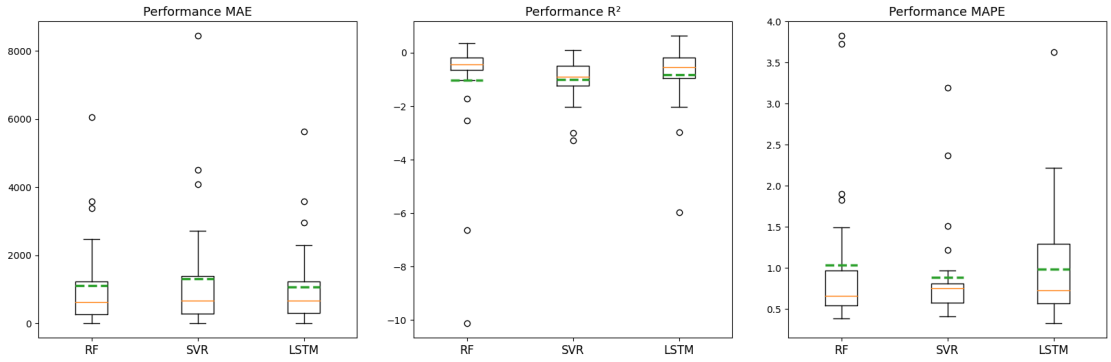
No geral, as visualizações confirmam as leituras das métricas. O LSTM se destaca pelas curvas suaves que produz, em contraste com os demais, o que está presente em todas as predições feitas e resultou em melhores predições por essa arquitetura. Estados em que a manifestação da doença não segue ciclos bem definidos apresentaram resultados insatisfatórios, e o contrário também foi verdade.

5.1.4 Impacto dos dados do Google Trends

Embora a bibliografia aponte a utilidade dos dados do Google Trends para modelos preditivos, foram feitos testes sem a presença desses números nos *datasets*, para analisar o seu real impacto para este trabalho. Novos modelos foram construídos e trei-

nados de acordo com o descrito na Seção 4.2, e após isso calcularam-se as métricas cuja distribuição pode ser vista em 5.10.

Figura 5.10 – Métricas para o teste sem a utilização de Google Trends. As linhas verdes tracejadas apresentam as médias; as linhas laranjas, as medianas.

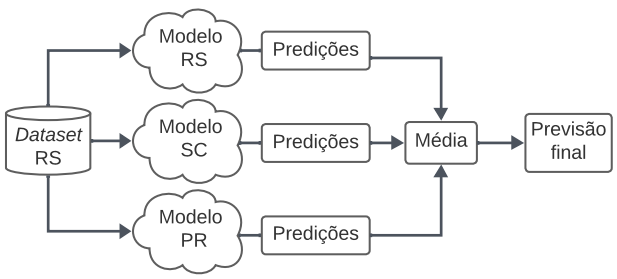


A diferença é pouca em relação aos modelos que utilizaram os dados, afinal trata-se de apenas um atributo, mas confirmou-se que a ausência tem impacto negativo. Todas as médias obtiveram leve piora com a retirada, e surgem *outliers* mais extremos, especialmente em MAPE e R².

5.2 *Ensemble* por macrorregião

Leituras sobre estados em isolado podem não cobrir o cenário completo da manifestação de uma doença, sendo importante também considerar fatores como a movimentação populacional e a existência de fenômenos climáticos semelhantes em territórios vizinhos. Pensando nisso, foi montado um método *ensemble* em que previsões para cada estado são feitas por todos os modelos da macrorregião em que ele está inserido.

Figura 5.11 – Exemplo de aplicação do *ensemble* por macrorregião.

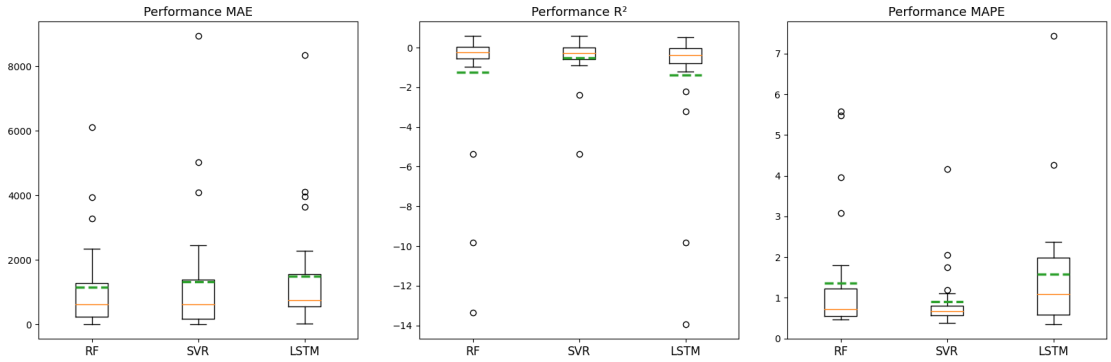


Fonte: o autor.

A ideia é, para cada estado, fazer previsões pelos modelos treinados sobre os demais estados da mesma macrorregião, com a previsão final sendo definida pela média desses valores. A Figura 5.11 contém um fluxograma exemplificando o processo aplicado ao estado do RS.

Variações desse processo foram testadas, com cada método em separado e com combinações diferentes entre eles. Como SVR e LSTM obtiveram performances similares nas predições individuais, acreditava-se que essa junção fosse a mais promissora, no entanto isso não se confirmou, com nenhuma combinação superando os resultados por método em isolado. São mostrados os *boxplots* das métricas por método em separado na imagem 5.12.

Figura 5.12 – Distribuição das métricas do *ensemble* por macrorregião. As linhas verdes tracejadas apresentam as médias; as linhas laranjas, as medianas.



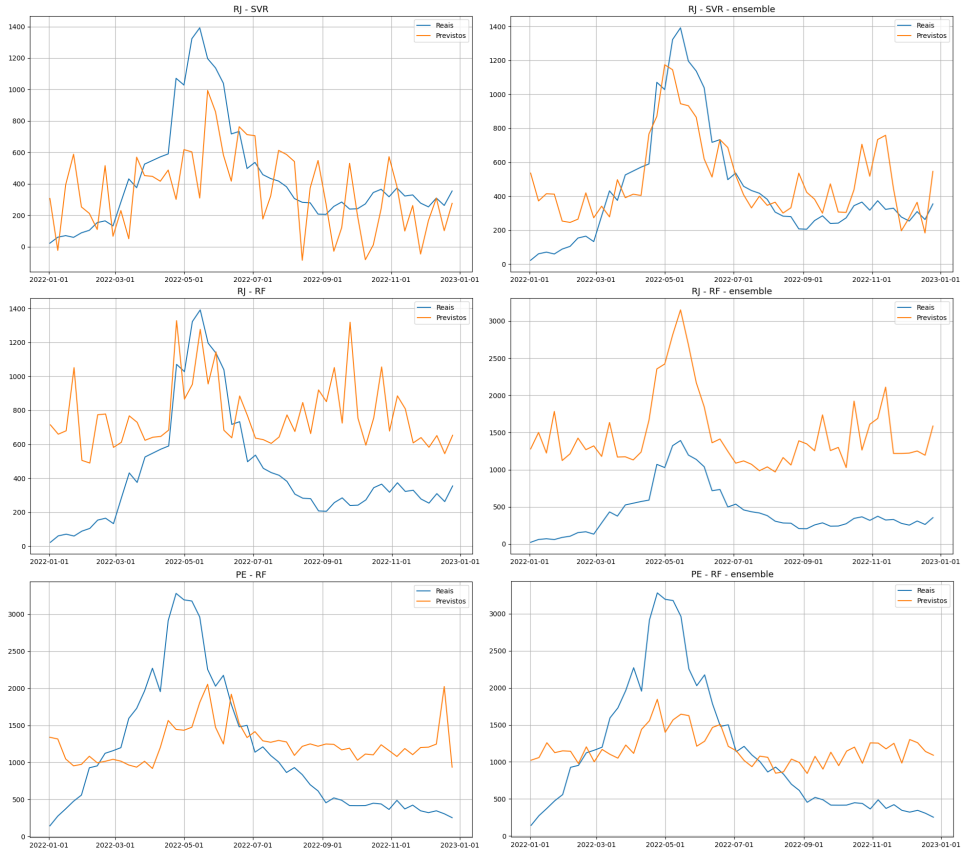
O processo acabou por resultar em leve diminuição de performance para os três métodos, especialmente pelas métricas MAPE e R^2. As previsões individuais haviam deixado claro que há grandes diferenças nas relações entre as variáveis de cada *dataset*, e esperava-se que através de *ensemble* um modelo pudesse suplementar outro, assim trazendo predições melhores. Diferente disso, os resultados indicam que ao agrupar os estados dessa forma, os modelos podem ter deixado de captar as peculiaridades de cada um deles.

A métrica que mais apresentou piora foi MAPE, com os *boxplots* para LSTM e RF exibindo concentração em faixa que ultrapassa os 200%. Para SVR a mudança não é tão gritante, mas nota-se aumento do extremo inferior, aproximando-se de 50%. Em R^2, os três métodos concentram seus valores mais a fundo na faixa dos negativos, mas a queda manifesta-se principalmente pelo surgimento de alguns *outliers* gritantes. Em MAE, pouca mudança foi observada, apenas leve aumento geral.

Algo que chamou atenção foram aqueles *outliers* com resultados extremos em determinadas métricas. As previsões de RF e LSTM sobre os *datasets* de Acre e Espírito Santo obtiveram R^2 incomumente baixos, e sobre Rio de Janeiro, Espírito Santo e Piauí obtiveram MAPEs demasiado altos. Com SVR, observou-se MAE bastante alto para o *dataset* de São Paulo. Reforça-se a observação obtida com as previsões individuais, de que alguns *datasets* apresentam padrões mais desafiadores que outros.

Assim como na Seção anterior, são separados alguns dos resultados de interesse, para apontar padrões detectados nas previsões como um todo. Esses exemplos são colocados ao lado das previsões originais, para que se possa compará-las. A Figura 5.13 contém exemplos do que se esperava que fosse acontecer com mais frequência neste experimento: uma suavização das curvas das previsões. Para o *dataset* do Rio de Janeiro isso aconteceu tanto com RF quanto com SVR, e para o de Pernambuco isso aconteceu com RF, em que o formato serrilhado das previsões originais foi reduzido para algo mais próximo de plano.

Figura 5.13 – Comparativo de previsões por *ensemble* para RJ e PE, com uso de RF e SVR. Na coluna da esquerda, as previsões originais; na da direita, aquelas feitas por *ensemble*.

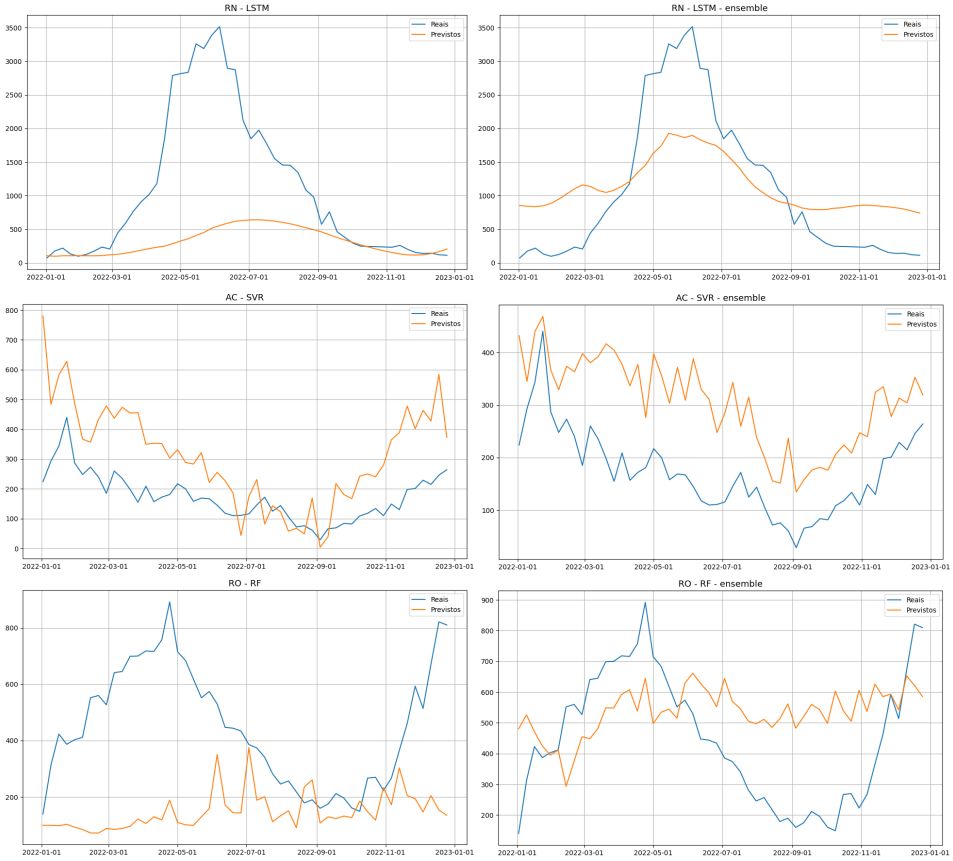


Fonte: o autor.

Suavizações do tipo podem surgir em *ensembles* quando há boa diversidade de arquiteturas entre os modelos, fazendo com que um complemente o outro, sem *outliers* extremos. Os ruídos presentes nas predições individuais cancelam-se mutuamente quando do cálculo da média, resultando em menos mudanças abruptas. É um padrão saudável, que no entanto pouco se manifestou nos resultados.

Os gráficos da Figura 5.14 contêm exemplos de um comportamento que foi visto com bastante frequência: os modelos previram valores mais altos do que anteriormente. Isso é bastante perceptível nos resultados obtidos por LSTM sobre o *dataset* de Rio Grande do Norte, e por RF sobre o de Rondônia. Os desenhos das curvas mantêm-se relativamente os mesmos, mas com números mais altos. Com a previsão de SVR sobre o Acre isso acontece apenas na parte central do conjunto de teste, com leve suavização nas extremidades.

Figura 5.14 – Comparativo de previsões por *ensemble* para RN, AC e RO, com uso dos três métodos. Na coluna da esquerda, as previsões originais; na da direita, aquelas feitas por *ensemble*.



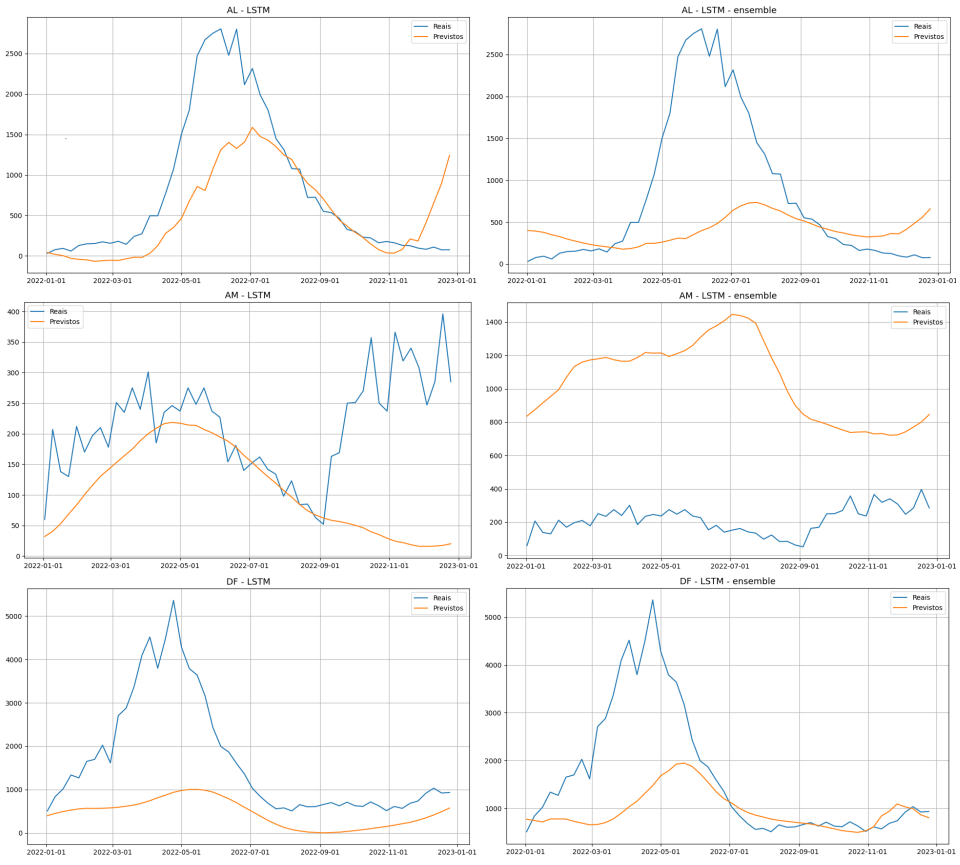
Fonte: o autor.

Se a suavização é sinal de uma complementaridade saudável, em que potenciais

outliers são negados pelas predições dos demais modelos, aqui pode estar acontecendo o oposto. É possível que algum modelo haja previsto valores muito altos e assim poluiu a previsão final. Ou ainda, que tenha havido consenso entre todos quanto à previsão de um padrão crescente nos dados, assim resultando em uma elevação da curva original.

Em geral as novas previsões feitas encaixam-se em um ou outro desses padrões, como reforçado pelo último conjunto de exemplos trazido em 5.15. Em Alagoas, a curva inicial foi achatada; no Amazonas, os modelos previram números muito maiores que na predição original; e no Distrito Federal também aconteceu aumento, mas que nesse caso acabou por aproximar as predições dos números reais.

Figura 5.15 – Comparativo de previsões por *ensemble* para AL, AM e DF, com uso de LSTM. Na coluna da esquerda, as previsões originais; na da direita, aquelas feitas por *ensemble*.



Fonte: o autor.

O experimento com o *ensemble* montado não resultou no que se esperava. Houve casos pontuais de leve melhora ou de mudanças negligíveis nas métricas selecionadas, mas em geral as diferenças de cada *dataset* parecem ter sido significativas o bastante para poluir as tentativas de previsão em conjunto, e as previsões individuais mostraram-se melhores.

6 CONCLUSÃO

Ao longo deste trabalho, foram montados modelos preditivos de casos de dengue para todas as unidades federativas do Brasil. O mesmo problema havia sido atacado em determinadas cidades no Brasil, mas tendo em vista experimentos de sucesso com dados estaduais em outros países, buscou-se explorar esse nível de concentração aqui também. Além de dados climáticos, cujo uso é consagrado nesse tipo de estudo, foi inserido nos *datasets* um atributo adicional contendo números do Google Trends, prática bem estabelecida em previsões de outras áreas. Finalmente, foi montado também um processo *ensemble*, acreditando que assim seria possível modelar de forma mais fiel o real cenário da manifestação da doença.

O experimento com os modelos individuais mostrou resultados mistos no geral. Um subconjunto dos *datasets* mostrou-se adequado para esse tipo de previsão, com leituras em nível aceitável conforme as métricas avaliativas escolhidas, e curvas de formatos que denotavam boa compreensão dos dados pelos modelos. Para outros *datasets*, porém, não foi possível atingir resultados aceitáveis. Esse cenário foi demonstrado através da listagem de todas as métricas, seguida por exemplos ilustrativos de padrões encontrados.

A montagem de um método *ensemble* por macrorregião trouxe resultados também mistos, com alterações pouco significativas mas pendendo para a piora nas métricas em geral. É possível que a manifestação da doença em diferentes estados de uma mesma macrorregião não siga padrões tão próximos, o que exigiria estudo mais aprofundado antes de se aproximar por esse ângulo novamente. Isso reforçou a noção de que esforços sobre este problema devem concentrar-se sobre aqueles estados específicos onde os dados possuam melhor qualidade.

De qualquer forma, o panorama geral que foi delineado demonstra que é possível montar um sistema preditivo com esse escopo no Brasil, contanto que se façam ajustes em relação ao que foi feito até aqui, e que o Google Trends é útil ao funcionamento de sistemas do tipo. Dentro disso, identificou-se também que métodos com base em LSTM e SVR ajustam-se bem à tarefa, que uma parcela das unidades federativas tem banco de dados climáticos robusto o bastante para previsões confiáveis, e que a adição de informações de motores de busca auxilia de fato nos resultados.

Trabalhos futuros devem debruçar-se com mais atenção sobre o pré-processamento dos dados, o que foi feito em nível relativamente baixo neste estudo. Experimentos com diferentes transformações e níveis de concentração poderiam trazer resultados drastica-

mente diferentes, suprindo lacunas que possam ter prejudicado o papel dos *datasets* em descrever a realidade de cada estado. Já que foi comprovada a utilidade do Google Trends, fica aberta a opção de explorá-lo mais a fundo, incluindo números de pesquisas com outros termos, e não somente com a palavra dengue como foi feito aqui. A agregação de dados diários e não semanais é outra alternativa possível, embora incompatível com o uso do Google Trends na forma como foi aplicado aqui.

Sugere-se, por fim, a diminuição do escopo para territórios menores, ou para unidades federativas específicas onde as avaliações aqui registradas foram mais positivas. Estados como Amapá, Amazonas, Bahia, Ceará, Minas Gerais, Mato Grosso, Mato Grosso do Sul, Pará e São Paulo, dentre outros que foram destacados anteriormente, indicam ser bons candidatos para isso. Os sistemas de monitoramento climático e epidemiológico do Brasil são muito bons, oferecendo a base necessária para esse tipo de estudo.

REFERÊNCIAS

AHMED, N. K. et al. An empirical comparison of machine learning models for time series forecasting. **Econometric reviews**, Taylor & Francis, v. 29, n. 5-6, p. 594–621, 2010.

AKIBA, T. et al. Optuna: A next-generation hyperparameter optimization framework. In: **Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining**. [S.l.: s.n.], 2019. p. 2623–2631.

ALPAYDIN, E. **Introduction to machine learning**. [S.l.]: MIT press, 2020.

ANTUNES, J. L. F.; CARDOSO, M. R. A. Uso da análise de séries temporais em estudos epidemiológicos. **Epidemiologia e Serviços de Saúde**, SciELO Brasil, v. 24, p. 565–576, 2015.

AWAD, M. et al. Support vector regression. **Efficient learning machines: Theories, concepts, and applications for engineers and system designers**, Springer, p. 67–80, 2015.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, p. 5–32, 2001.

BROCKWELL, P. J.; DAVIS, R. A. **Time series: theory and methods**. [S.l.]: Springer science & business media, 2009.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. [S.l.: s.n.], 2016. p. 785–794.

CODEÇO, C. T. et al. Infodengue: a nowcasting system for the surveillance of dengue fever transmission. **BioRxiv**, Cold Spring Harbor Laboratory, p. 046193, 2016.

ELITH, J.; LEATHWICK, J. R.; HASTIE, T. A working guide to boosted regression trees. **Journal of animal ecology**, Wiley Online Library, v. 77, n. 4, p. 802–813, 2008.

FACELI, K. et al. Inteligência artificial: uma abordagem de aprendizado de máquina. 2021.

FATHI, O. Time series forecasting using a hybrid arima and lstm model. **Velvet Consulting**, p. 1–7, 2019.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). **The annals of statistics**, Institute of Mathematical Statistics, v. 28, n. 2, p. 337–407, 2000.

GINSBERG, J. et al. Detecting influenza epidemics using search engine query data. **Nature**, Nature Publishing Group UK London, v. 457, n. 7232, p. 1012–1014, 2009.

GUNN, S. R. et al. Support vector machines for classification and regression. **ISIS technical report**, v. 14, n. 1, p. 5–16, 1998.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997.

HOTA, H.; HANDA, R.; SHRIVAS, A. K. Time series data prediction using sliding window based rbf neural network. **International Journal of Computational Intelligence Research**, v. 13, n. 5, p. 1145–1156, 2017.

HOYOS, W.; AGUILAR, J.; TORO, M. Dengue models based on machine learning techniques: A systematic literature review. **Artificial intelligence in medicine**, Elsevier, v. 119, p. 102157, 2021.

JUN, S.-P.; YOO, H. S.; CHOI, S. Ten years of research change using google trends: From the perspective of big data utilizations and applications. **Technological forecasting and social change**, Elsevier, v. 130, p. 69–87, 2018.

JUNIOR, J. B. S. et al. Epidemiology and costs of dengue in brazil: a systematic literature review. **International Journal of Infectious Diseases**, v. 122, p. 521–528, 2022. ISSN 1201-9712. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1201971222003836>.

KAHRAMAN, A. et al. Comparison of the effect of regularization techniques and lookback window length on deep learning models in short term load forecasting. In: SPRINGER. **The Purple Mountain Forum on Smart Grid Protection and Control**. [S.l.], 2019. p. 655–669.

KOPARANOV, K. A.; GEORGIEV, K. K.; SHTEREV, V. A. Lookback period, epochs and hidden states effect on time series prediction using a lstm based neural network. In: IEEE. **2020 28th National Conference with International Participation (TELECOM)**. [S.l.], 2020. p. 61–64.

LAHIRI, S.; GHANTA, K. C. The support vector regression with the parameter tuning assisted by a differential evolution technique: Study of the critical velocity of a slurry flow in a pipeline. **Chemical Industry and Chemical Engineering Quarterly/CICEQ**, v. 14, n. 3, p. 191–203, 2008.

LAINDER, A. D.; WOLFINGER, R. D. Forecasting with gradient boosted trees: augmentation, tuning, and cross-validation strategies: Winning solution to the m5 uncertainty competition. **International Journal of Forecasting**, Elsevier, v. 38, n. 4, p. 1426–1433, 2022.

LI, J. Assessing the accuracy of predictive models for numerical data: Not r nor r2, why not? then what? **PloS one**, Public Library of Science San Francisco, CA USA, v. 12, n. 8, p. e0183250, 2017.

LIAW, A.; WIENER, M. et al. Classification and regression by randomforest. **R news**, v. 2, n. 3, p. 18–22, 2002.

LOURENÇO, J. et al. Challenges in dengue research: A computational perspective. **Evolutionary applications**, Wiley Online Library, v. 11, n. 4, p. 516–533, 2018.

MAJEED, M. A. et al. A deep learning approach for dengue fever prediction in malaysia using lstm with spatial attention. **International journal of environmental research and public health**, MDPI, v. 20, n. 5, p. 4130, 2023.

MAKRIDAKIS, S.; SPILIOTIS, E.; ASSIMAKOPOULOS, V. Statistical and machine learning forecasting methods: Concerns and ways forward. **PloS one**, Public Library of Science San Francisco, CA USA, v. 13, n. 3, p. e0194889, 2018.

MAKRIDAKIS, S.; SPILIOTIS, E.; ASSIMAKOPOULOS, V. The m4 competition: 100,000 time series and 61 forecasting methods. **International Journal of Forecasting**, Elsevier, v. 36, n. 1, p. 54–74, 2020.

MAKRIDAKIS, S.; SPILIOTIS, E.; ASSIMAKOPOULOS, V. M5 accuracy competition: Results, findings, and conclusions. **International Journal of Forecasting**, Elsevier, v. 38, n. 4, p. 1346–1364, 2022.

MAKRIDAKIS, S.; SPILIOTIS, E.; ASSIMAKOPOULOS, V. The m5 competition: Background, organization, and implementation. **International Journal of Forecasting**, Elsevier, v. 38, n. 4, p. 1325–1336, 2022.

MARQUES-TOLEDO, C. d. A. et al. Dengue prediction by the web: Tweets are a useful tool for estimating and forecasting dengue at country and city level. **PLoS neglected tropical diseases**, Public Library of Science San Francisco, CA USA, v. 11, n. 7, p. e0005729, 2017.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning**. [S.l.]: MIT press, 2018.

MUSSUMECI, E.; COELHO, F. C. Large-scale multivariate forecasting models for dengue-lstm versus random forest regression. **Spatial and Spatio-temporal Epidemiology**, Elsevier, v. 35, p. 100372, 2020.

NETO, A. C. L. et al. A incidência de dengue no brasil, pós pandemia covid-19: redução do número de casos ou aumento de subnotificações? uma revisão integrative. **Brazilian Journal of Health Review**, v. 6, n. 1, p. 3010–3021, 2023.

NGUYEN, V.-H. et al. Deep learning models for forecasting dengue fever based on climate data in vietnam. **PLoS Neglected Tropical Diseases**, Public Library of Science San Francisco, CA USA, v. 16, n. 6, p. e0010509, 2022.

NGUYEN, X. H. et al. Combining statistical machine learning models with arima for water level forecasting: The case of the red river. **Advances in Water Resources**, Elsevier, v. 142, p. 103656, 2020.

NOBLE, W. S. What is a support vector machine? **Nature biotechnology**, Nature Publishing Group UK London, v. 24, n. 12, p. 1565–1567, 2006.

ONYUTHA, C. From r-squared to coefficient of model accuracy for assessing"goodness-of-fits". **Geoscientific Model Development Discussions**, Copernicus GmbH, p. 1–25, 2020.

PASCANU, R.; MIKOLOV, T.; BENGIO, Y. On the difficulty of training recurrent neural networks. In: PMLR. **International conference on machine learning**. [S.l.], 2013. p. 1310–1318.

PLIEGO, E. P.; VELÁZQUEZ-CASTRO, J.; COLLAR, A. F. Seasonality on the life cycle of aedes aegypti mosquito and its statistical relation with dengue outbreaks. **Applied Mathematical Modelling**, Elsevier, v. 50, p. 484–496, 2017.

SIAMI-NAMINI, S.; TAVAKOLI, N.; NAMIN, A. S. A comparison of arima and lstm in forecasting time series. In: IEEE. **2018 17th IEEE international conference on machine learning and applications (ICMLA)**. [S.l.], 2018. p. 1394–1401.

SILVA, I. S. et al. Observatório da dengue: surveillance based on twitter sentiment stream analysis. In: **Proceedings of the Brazilian Symposium on Databases, Demos Track. Florianópolis, Brazil**. [S.l.: s.n.], 2011. p. 49–54.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

STOLERMAN, L. M.; MAIA, P. D.; KUTZ, J. N. Forecasting dengue fever in brazil: An assessment of climate conditions. **PloS one**, Public Library of Science San Francisco, CA USA, v. 14, n. 8, p. e0220106, 2019.

ZHAO, N. et al. Machine learning and dengue forecasting: Comparing random forests and artificial neural networks for predicting dengue burden at national and sub-national scales in colombia. **PLOS Neglected Tropical Diseases**, Public Library of Science San Francisco, CA USA, v. 14, n. 9, p. e0008056, 2020.