



## REDES NEURAIIS RECORRENTES PARA PREVISÃO DA QUALIDADE DO AR NO MUNICÍPIO DO RIO DE JANEIRO

José Fernando Lopes Leocadio

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do grau de Mestre em Engenharia Civil.

Orientador: Nelson Francisco Favilla Ebecken

Rio de Janeiro  
Fevereiro de 2023

REDES NEURAIIS RECORRENTES PARA PREVISÃO DA QUALIDADE DO AR  
NO MUNICÍPIO DO RIO DE JANEIRO

José Fernando Lopes Leocadio

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO  
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA  
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE  
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE  
EM ENGENHARIA CIVIL.

Examinada por:

---

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

---

Prof. Rogério Pinto Espíndola, D.Sc.

---

Prof. Mario Antonio Ribeiro Dantas, Ph.D.

RIO DE JANEIRO, RJ BRASIL

FEVEREIRO DE 2023

Leocadio, José Fernando Lopes.

Redes Neurais Recorrentes para a previsão da qualidade do ar no município do Rio de Janeiro/ José Fernando Lopes Leocadio. – Rio de Janeiro: UFRJ/COPPE, 2023.

xv, 76p.: il.; 29,7 cm.

Orientador: Nelson Francisco Favilla Ebecken

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia Civil, 2022.

Referências Bibliográficas: p. 72-76.

1. Qualidade do Ar. 2. Rede Neural Recorrente. 3. Sistemas Computacionais. 4. Engenharia Ambiental I. Ebecken, Nelson Francisco Favilla *et al.*. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

## **Agradecimentos**

Ao Instituto Alberto Luiz Coimbra de Pós-Graduação de Pesquisa de Engenharia, por ter acreditado em meu potencial.

Ao meu orientador, Nelson Favilla Ebecken, pelos ensinamentos e pelos auxílios prestados durante todo o mestrado.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pela indispensável ajuda financeira no início do curso.

À minha mãe, Geralda Lopes Leocadio, eternamente minha maior motivadora.

Ao meu pai, José Leocadio, a pessoa mais disciplinada e trabalhadora a qual conheço, que me serve de modelo até hoje.

À Deus, que sempre esteve ao meu lado nas horas mais difíceis.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## REDES NEURAIS RECORRENTES PARA PREVISÃO DA QUALIDADE DO AR NO MUNICÍPIO DO RIO DE JANEIRO

José Fernando Lopes Leocadio

Fevereiro/2023

Orientadores: Nelson Francisco Favilla Ebecken

Programa: Engenharia Civil

O clima tropical da região metropolitana do Rio de Janeiro é especialmente suscetível a poluentes atmosféricos, diretamente relacionados a graves doenças cardiopulmonares. Como forma de contribuir com a resolução deste problema, neste trabalho pretendemos examinar o desempenho de um *ensemble* de redes neurais do tipo recorrente, na tarefa de previsão dos níveis máximos de poluentes atmosféricos. O conjunto de dados analisado, disponibilizado pelo governo municipal, é composto pelas variações dos níveis de poluentes atmosféricos e por informações meteorológicas, de oito bairros distintos do Rio de Janeiro. Verificou-se que a correlação entre os dados de estações adjacentes pode chegar a 95%, de acordo com a variável estudada. Os experimentos realizados mostraram que o *ensemble* de redes neurais proposto, tem desempenho superior aos modelos simples, em 3 dos 4 cenários estudados. Adicionalmente, projetamos um sistema de alerta para níveis de poluentes anormais, utilizando ferramentas de computação em nuvem.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

RECURRENT NEURAL NETWORKS FOR AIR-QUALITY FORECAST MODELS  
IN THE CITY OF RIO DE JANEIRO

José Fernando Lopes Leocadio

February/2023

Advisors: Nelson Francisco Favilla Ebecken

Department: Civil Engineering

The tropical climate of the metropolitan region of Rio de Janeiro is especially susceptible to air pollutants, that could be directly connected to serious cardiopulmonary illnesses. To contribute with the resolution of this issue, the goal of the present work is to exam the performance of an ensemble of Recurrent Neural Networks on the prediction of daily maximum pollutant levels. The analyzed dataset is provided by the local government and it is composed by hourly-levels for pollutants and meteorological features from eight different locations of Rio de Janeiro. We verified that the correlation among nearby stations data is up to 95%, depending on the examined variable. The experiments showed that the ensemble model has superior performance to simpler models in 3 out of 4 studied scenarios. Additionally, we designed an alert system to abnormal pollutant levels, using cloud computing tools.

# Sumário

1	Introdução.....	1
1.1	Objetivos gerais .....	2
1.2	Objetivos específicos.....	3
2	Redes neurais para a previsão da qualidade do ar .....	4
2.1	Redes neurais recorrentes .....	4
2.1.1	O funcionamento das redes neurais recorrentes .....	4
2.1.2	Aprendizado nas redes neurais recorrentes .....	5
2.2	Long-Short Term Memory .....	7
2.2.1	O funcionamento da rede LSTM.....	7
2.2.2	Redes neurais LSTM: equacionamento.....	8
2.2.3	LSTM bidirecional .....	10
2.3	Gated Recurrent Unity .....	10
2.3.1	O funcionamento da rede GRU .....	10
2.3.2	Portão de redefinição.....	12
2.3.3	Portão de atualização.....	13
2.3.4	Combinação de resultados.....	13
3	Metodologia.....	15
3.1	Método de preenchimento de valores ausentes .....	15
3.2	Escalonamento dos dados .....	16
3.3	Reframe do problema .....	16
3.4	Número de neurônios iniciais .....	17
3.5	Tamanho do input de dados (LAG).....	17
3.6	Múltiplo input e múltiplo output .....	18
3.7	Validação cruzada.....	19

3.8 Testes preliminares .....	20
3.9 Ensemble de modelos .....	22
4 Estudo de caso: qualidade do ar no município do Rio .....	25
4.1 Contextualização.....	25
4.2 MonitorAr-Rio.....	26
4.3 Variáveis meteorológicas e poluentes atmosféricos .....	28
4.3.1 Temperatura.....	28
4.3.2 Pressão atmosférica .....	29
4.3.3 Velocidade do vento .....	29
4.3.4 Chuva.....	30
4.3.5 Umidade relativa do ar .....	30
4.3.6 Óxidos de nitrogênio .....	31
4.3.7 Monóxido de carbono.....	31
4.3.8 Enxofre .....	32
4.3.9 Material particulado.....	32
4.3.10 Ozônio atmosférico .....	33
4.4 Arquitetura sugerida para o modelo em produção.....	33
5 Análise exploratória dos dados.....	36
5.1 Descrição dos dados .....	36
5.1 Resumo estatístico .....	37
5.2 Estado do preenchimento do conjunto de dados .....	39
5.3 Correlação entre dados de estações adjacentes.....	43
5.3.1 Chuva.....	44
5.3.2 Pressão atmosférica .....	45
5.3.3 Radiação solar .....	46

5.3.4	Temperatura.....	47
5.3.5	Umidade relativa .....	48
5.3.6	Velocidade do vento .....	49
5.3.7	Dióxido de enxofre (SO <sub>2</sub> ).....	50
5.3.8	Monóxido de carbono (CO).....	51
5.3.9	Ozônio (O <sub>3</sub> ) .....	52
5.3.10	Material Particulado (PM <sub>10</sub> ) .....	53
6	Resultados e discussão.....	54
6.1	Testes preliminares .....	54
6.1.1	Tamanho do batch de dados .....	55
6.1.2	Arquitetura da rede neural .....	56
6.1.3	Número de nós na camada intermediária .....	58
6.1.4	Duração do treinamento .....	59
6.1.5	Número de camadas intermediárias.....	59
6.1.6	Normalização de batch e dropout .....	61
6.2	Ensemble de modelos .....	63
6.2.1	São Cristóvão.....	63
6.2.2	Tijuca.....	64
6.2.3	Copacabana.....	65
6.2.4	Centro .....	66
7	Conclusão e trabalhos futuros.....	69
8	Referências Bibliográficas.....	72

## LISTA DE FIGURAS

Figura 1 unidade LSTM <sup>[21]</sup> .....	7
Figura 2 estrutura da rede LSTM bidirecional .....	10
Figura 3 comparação entre GRU e LSTM .....	11
Figura 4 estrutura da unidade GRU <sup>[25]</sup> .....	12
Figura 5 validação cruzada em dados não-sequenciais .....	19
Figura 6 validação cruzada em séries temporais .....	20
Figura 7 testes realizados para definir melhor conjunto de hiperparâmetros .....	21
Figura 8 esquema do ensemble dos resultados das redes neurais.....	24
Figura 9 localização das estações do Monitorar-Rio .....	26
Figura 10 funcionamento da rede de captação de dados .....	27
Figura 11 padrões de qualidade do ar do CONAMA .....	28
Figura 12 arquitetura sugerida para a aplicação em produção .....	34
Figura 13 grau de preenchimento dos dados da estação de Bangu .....	39
Figura 14 grau de preenchimento da estação do Centro.....	40
Figura 15 grau de preenchimento da estação de Campo Grande .....	40
Figura 16 grau de preenchimento da estação do Copacabana .....	41
Figura 17 grau de preenchimento da estação do Irajá .....	41
Figura 18 grau de preenchimento da estação de Pedra de Guaratiba .....	42
Figura 19 grau de preenchimento da estação de São Cristóvão .....	42
Figura 20 grau de preenchimento da estação da Tijuca.....	43
Figura 21 correlação entre a chuva nas oito estações.....	44
Figura 22 correlação entre a pressão atmosférica nas oito estações.....	45
Figura 23 correlação entre a radiação solar nas oito estações .....	46
Figura 24 correlação entre a temperatura nas oito estações .....	47

Figura 25 correlação entre a umidade relativa nas oito estações.....	48
Figura 26 correlação entre a velocidade do vento nas oito estações .....	49
Figura 27 correlação entre o SO <sub>2</sub> nas oito estações.....	50
Figura 28 correlação entre o CO nas oito estações.....	51
Figura 29 correlação entre o O <sub>3</sub> nas oito estações .....	52
Figura 30 correlação entre o PM <sub>10</sub> nas oito estações.....	53

## **LISTA DE TABELAS**

Tabela 1 atributos do conjunto de dados de monitoramento de poluição atmosférica...	37
Tabela 2 Resumo estatístico dos dados da estação do Centro.....	38
Tabela 3 Resumo estatístico dos dados da estação de Copacabana .....	38
Tabela 4 Resumo estatístico dos dados da estação de São Cristóvão .....	38
Tabela 5 Resumo estatístico dos dados da estação da Tijuca.....	39
Tabela 6 resumo dos resultados obtidos .....	55
Tabela 7 comparação do desempenho do modelo ensemble com o desempenho das redes simples, utilizando dados da estação de São Cristóvão .....	63
Tabela 8 comparação do desempenho do modelo ensemble com o desempenho das redes simples, utilizando dados da estação da Tijuca.....	64
Tabela 9 comparação do desempenho do modelo ensemble com o desempenho das redes simples, utilizando dados da estação de Copacabana .....	65
Tabela 10 comparação do desempenho do modelo ensemble com o desempenho das redes simples, utilizando dados da estação do Centro.....	66

## **LISTA DE ABREVIATURAS**

ADAM- Adaptive Moment Estimation

BI- LSTM- Bidirectional Long Short Term Memory

BPTT- Back Propagation Through Time

CNN- Convolutional Neural Network

CONAMA- Conselho Nacional do Meio Ambiente

GRU- Gated Recurrent Unity

LSTM- Long-Short Term Memory

MAE- Mean Absolute Error

OMS – Organização Mundial da Saúde

RMSE- Root Mean Square Error

RNA- Rede Neural Artificial

RNN- Recurrent Neural Network

SVM- Suport Vector Machine

# 1 Introdução

De acordo com a Organização Mundial da Saúde (OMS) <sup>[1]</sup> desde 2018, 4.1 milhões de pessoas morreram por causas relacionadas à poluição atmosférica. 91% destas mortes ocorreram em países de terceiro mundo, como o Brasil. O rápido desenvolvimento industrial destes países geralmente é associado a grandes mudanças climáticas, o que agrava ainda mais a situação. Neste cenário, o clima tropical da região metropolitana do Rio de Janeiro parece ser especialmente suscetível a poluentes atmosféricos, que podem ser diretamente ligados a doenças cardiopulmonares.

Previamente Ghoneim *et al.* <sup>[2]</sup>, Luna *et al.* <sup>[3]</sup> e Li *et al.* <sup>[4]</sup> testaram a performance de modelos clássicos como as redes neurais do tipo *Feed-Forward* e de Máquinas de Vetor de Suporte (SVM) na tarefa de previsão dos níveis horários de ozônio e material particulado, obtendo resultados razoáveis. Entretanto, devido à natureza não linear dos dados meteorológicos, métodos estatísticos clássicos não possuem um bom desempenho na tarefa mencionada.

Com o intuito de aprender as especificidades contidas nos dados de qualidade do ar, a literatura recomenda a utilização de modelos mais robustos, como as Redes Neurais Recorrentes (RNN). Kok *et al.* <sup>[5]</sup>, Fan *et al.* <sup>[6]</sup>, Li *et al.* <sup>[7]</sup>, Bui *et al.* <sup>[8]</sup>, Sak *et al.* <sup>[9]</sup>, Pardo e Malpica <sup>[10]</sup>, Wang *et al.* <sup>[11]</sup>, Navares e Aznarte <sup>[12]</sup> utilizaram *Long-Short Term Memory* (LSTM) na tarefa de previsão de níveis horários de poluentes, em diferentes cidades pelo mundo. Esta variante das RNN tem a capacidade de aprender dependências de longo termo em séries temporais e já mostrou resultados promissores em diversos campos. Todavia, quando utilizada em conjunto com dados de qualidade do ar, a rede LSTM tem desempenho ainda aquém do esperado.

Uma vez que previsões de curto prazo não têm aplicação na prática, Hossain *et al.* <sup>[13]</sup> propôs a utilização de redes RNN e suas variantes para a previsão dos níveis máximos diários para cada poluente. Para melhorar a precisão dos resultados fornecidos pela RNN, Du *et al.* <sup>[14]</sup> e Zhang *et al.* <sup>[15]</sup> sugeriram o ensemble de LSTM e Redes Neurais Convolucionais (CNN), de forma que cada rede neural ficou encarregada de aprender diferentes nuances dos dados.

Neste cenário, o presente trabalho trouxe uma proposta de construção de um modelo ensemble, utilizando três diferentes arquiteturas de RNN, para a tarefa de previsão dos níveis máximos diários de poluentes. Supostamente, nos trabalhos supracitados, foi necessário treinar um modelo para a previsão de cada variável separadamente, o que poderia ser inviável para grandes conjuntos de dados. Para solucionar esta questão e para melhor entendimento do desempenho do modelo proposto, este projeto utilizou a técnica de múltiplo-*input* e múltiplo *output*, de forma que o modelo forneceu previsões para todas as variáveis de uma só vez.

Este trabalho está estruturado da seguinte forma: no capítulo 2 foi feita uma breve introdução sobre a utilização de redes neurais na previsão de níveis de poluição atmosférica. A seguir, neste mesmo capítulo, foi feita uma explanação sobre a matemática por trás das redes neurais recorrentes. No capítulo 3, foi explicada a metodologia utilizada para o tratamento de dados e para a montagem e treinamento dos modelos utilizados neste trabalho. No capítulo 4, foi explicado o estudo de caso deste trabalho, que utilizou as redes neurais citadas no capítulo 2, para a previsão da qualidade do ar no município do Rio de Janeiro. No capítulo 5, foram exibidos os resultados da análise exploratória prévia dos dados utilizados no projeto. No capítulo 6 foram expostos e comentados os resultados fornecidos por cada modelo de previsão, utilizado no trabalho. O capítulo 7 mostra as conclusões obtidas após as análises dos resultados e as orientações para trabalhos futuros na mesma linha.

## **1.1 Objetivos gerais**

- Estudar como a variação climática e meteorológica influencia na qualidade do ar, avaliando a evolução dos níveis de poluentes atmosféricos, ao longo do tempo;
- Utilizando técnicas de aprendizado de máquina, desenvolver um modelo robusto, que possa ser aplicado a problemas que envolvam séries temporais;

## **1.2 Objetivos específicos**

- Desenvolver um modelo que alerte quando houver possibilidade de níveis de poluentes fora do normal;
- Estudar a correlação entre os dados de estações adjacentes, e possível utilização desta correlação no desenvolvimento de um método para preencher valores ausentes;
- Estudar se o ensemble de algoritmos obtém um desempenho melhor que os algoritmos simples, na previsão da qualidade atmosférica;

## 2 Redes neurais para a previsão da qualidade do ar

Grossi *et al* <sup>[16]</sup> define as redes neurais artificiais (RNA) como sistemas adaptativos, inspirados no funcionamento dos neurônios biológicos, que são capazes de modificar sua estrutura interna, de acordo com uma função objetivo. O mesmo autor afirma também que versões mais modernas deste algoritmo são particularmente adequadas para a resolução de problemas não lineares, como os que envolvem dados de poluição atmosférica.

Um exemplo da eficácia das redes neurais foi dado por Fontes *et al* <sup>[17]</sup>, que construiu um modelo capaz de identificar a origem da emissão dos poluentes atmosféricos com alta acurácia, na cidade de Madri.

Num esforço similar ao feito no trabalho de Fontes *et al*, Ibrahim *et al* <sup>[18]</sup> utilizou uma versão moderna das redes neurais recorrentes, a *Long Short Term Memory* (LSTM), obtendo bons resultados. Neste trabalho Ibrahim afirma ainda que na medida em que as séries temporais de dados meteorológicos costumam conter muitos ruídos, é encontrado um cenário ideal para a utilização de redes neurais.

A eficácia de uma rede neural geralmente está atrelada a um ajuste correto de seus hiperparâmetros. Por esta razão, Xian Li *et al* <sup>[19]</sup> e Pardo *et al* <sup>[20]</sup> afirmam que o desempenho das redes neurais deve ainda ser visto com ressalvas. Todavia, Pardo *et al* aponta que, com a configuração adequada, os modelos de rede recorrentes foram capazes de superar, com menor custo computacional, o desempenho de avançados sistemas meteorológicos.

A seguir é feita uma explanação teórica a respeito das redes neurais recorrentes e suas variantes, utilizadas neste trabalho.

### 2.1 Redes neurais recorrentes

#### 2.1.1 O funcionamento das redes neurais recorrentes

Redes neurais recorrentes (RNN) são redes neurais específicas para reconhecimento de padrões em sequências de dados. A principal diferença estrutural

entre as RNN e as redes neurais clássicas é a presença de um loop de *feedback*, que é conectado à rede, desta forma ingerindo as próprias saídas como entrada, em etapas posteriores do treinamento.

Pode ser dito então que, as RNN têm duas fontes de *input*, a entrada de dados atual e as informações que são retidas ao longo do tempo, que são combinadas para determinar a entrada final do modelo.

A informação sequencial é preservada no estado oculto da rede. O processo de carregamento da memória da rede RNN adiante pode ser descrito matematicamente como:

$$h_t = \phi(Wx_t + Ux_{t-1}) \quad (1)$$

Onde:

$h_t$ : estado oculto na etapa de tempo t

$x_t$ : entrada no tempo t

$W$ : matriz de peso, adicionada ao estado oculto da etapa anterior  $h_{t-1}$  multiplicada pelo seu próprio estado oculto.

$U$ : matriz de transição com pesos do estado anterior

Nesta equação, a soma das matrizes de peso e do estado oculto é multiplicada pela função  $\phi$  ou por uma função sigmoide/tangente hiperbólica, para que valores muito altos ou baixos sejam encaixados num mesmo espaço logístico.

O algoritmo utilizado na otimização dos pesos e redistribuição dos erros entre as conexões das redes neurais recorrentes é o *Back Propagation Through Time* (BPTT), que é explicado a seguir.

### **2.1.2 Aprendizado nas redes neurais recorrentes**

Sejam as equações básicas das redes recorrentes:

$$s_t = \tanh(Ux_t + Ws_{t-1}) \quad (2)$$

$$\hat{y}_t = \text{softmax}(Vs_t) \quad (3)$$

Onde,  $s_t$  é o estado oculto no tempo  $t$ , ou memória da rede.  $\hat{y}_t$  é a saída da rede no tempo  $t$  e  $Vs_t$  o vetor de previsões do modelo.

O erro do treinamento é representado por:

$$E_t(y_t, \hat{y}_t) = -y_t \log \hat{y}_t \quad (4)$$

$$E(y, \hat{y}) = \sum_t E_t(y_t, \hat{y}_t) = - \sum_t y_t \log \hat{y}_t \quad (5)$$

Onde  $y_t$  é o valor real e  $\hat{y}_t$  é o valor previsto pelo modelo. O erro total é representado como soma dos erros de cada etapa do treino. De forma semelhante, são somados os gradientes em cada etapa de tempo do treinamento:

$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W} \quad (6)$$

Os gradientes são calculados com a regra de diferenciação da cadeia. Para  $E_3$  por exemplo:

$$\frac{\partial E_3}{\partial V} = \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial V} = \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial z_3} \frac{\partial z_3}{\partial V} = (\hat{y}_3 - y_3) \otimes s_3 \quad (7)$$

Onde  $z_3 = Vs_3$  e a última parcela representa o produto externo de dois vetores. Da relação acima, é possível deduzir que  $\frac{\partial E_3}{\partial V}$  depende apenas dos valores atuais  $\hat{y}_3$ ,  $y_3$  e  $s_3$ .

Para o cálculo de  $W$ , a regra da cadeia acima pode ser escrita como:

$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial W} \quad (8)$$

$$s_3 = \tanh(Ux_t + Ws_{t-1}) \quad (9)$$

Uma vez que a equação (9) depende de  $s_2$ , que depende de  $W$  e  $s_1$  e assim por diante, a equação do BPTT, considerando  $t=3$  a  $t=0$  pode ser escrita da seguinte forma:

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W} \quad (10)$$

## 2.2 Long-Short Term Memory

### 2.2.1 O funcionamento da rede LSTM

Uma das variações da arquitetura da rede neural recorrente, as redes LSTM são especialmente ajustadas para a construção de modelos de previsão em séries temporais, que possuem intervalos de tempo de duração desconhecida.

Estruturalmente, uma unidade da rede LSTM, mostrada na figura 1, é composta de uma célula, uma porta de *input*, uma porta de *output* e uma porta de esquecimento. A célula tem como finalidade reter as informações captadas pelo modelo, enquanto os portões realizam manipulações de memória.

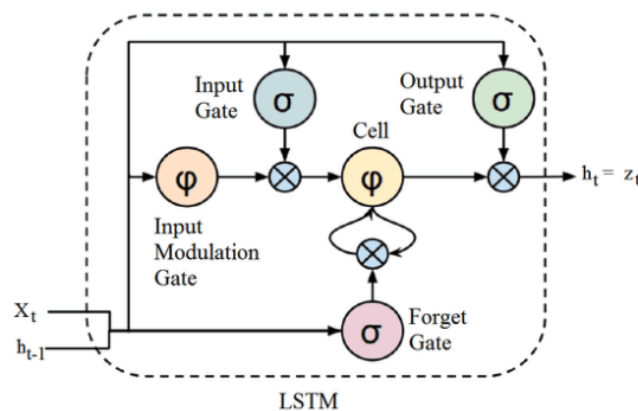


Figura 1 unidade LSTM <sup>[21]</sup>

A porta de esquecimento é a parte da unidade LSTM que faz a remoção de informações que não são mais úteis no estado da célula. Esta porta recebe dois *inputs*:  $x_t$  (entrada do momento específico) e  $h_{t-1}$  (saída da célula anterior). Estes dois *inputs* são multiplicados pela matriz de pesos e logo após, adicionados à parcela de *bias*. A seguir, o resultado desta soma passa por uma função de ativação, usualmente a função sigmóide, que fornecerá uma saída binária. Se a saída for 0, a informação é esquecida, e caso a saída seja 1, a informação é retida para uso futuro.

A adição de informações ao estado da célula é feita através da porta de *input*. Num primeiro momento, a informação é regularizada, usando a função sigmoide, que de forma similar a da porta de esquecimento, vai filtrar os valores a serem lembrados utilizando  $x_t$  e  $h_{t-1}$ . A seguir, um vetor é criado, utilizando a função tangente hiperbólica, que tem como saída valores de -1 a 1. Este vetor conterà todos os valores possíveis para  $x_t$  e  $h_{t-1}$ . Os valores deste vetor e os valores regularizados são multiplicados, para obter informações úteis ao modelo.

A última porta, de *output*, possui a função de extrair a informação útil do estado atual da célula e apresentá-la como saída. Primeiramente, é gerado um vetor, aplicando-se a função tangente hiperbólica à célula. Após isto, a informação é regularizada com a função sigmoide, que vai filtrar os valores a serem lembrados, usando as entradas  $h_{t-1}$  e  $x_t$ . Os valores do vetor e os valores regularizados são então multiplicados, para serem enviados como saída e entrada, para a próxima célula.

### 2.2.2 Redes neurais LSTM: equacionamento

Nas equações abaixo, as variáveis em letras minúsculas, representam vetores. Matrizes  $W_q$  e  $U_q$  contêm respectivamente, os pesos do *input* e as conexões recorrentes, onde o subscrito q pode significar tanto a porta de *input* i, *output* o, ou porta de esquecimento f da célula de memória c, dependendo da ativação sendo calculada.

Nestas equações, é utilizada a ‘notação de vetor’. Então, por exemplo,  $c_t \in R^h$  não é somente uma célula de uma das unidades LSTM, mas contém h unidades da referida célula. As equações compactas da passagem de dados em uma unidade LSTM são:

$$f_t = \sigma_g(W_t x_t + U_f h_{t-1} + b_f) \quad (11)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (12)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \quad (13)$$

$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \quad (14)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \quad (15)$$

Onde os valores iniciais são  $c_0 = 0$  e  $h_0 = 0$  e o operador  $\circ$  denota o produto de Hadamard. O subscrito  $t$  indexa o passo de acordo com o tempo. As variáveis presentes nestas equações são identificadas a seguir:

- $x_t \in R^d$ : vetor input para a unidade LSTM;
- $f_t \in R^h$ : vetor de ativação da porta de esquecimento;
- $i_t \in R^h$ : vetor de ativação da porta de *input/update*;
- $o_t \in R^h$ : vetor de ativação da porta de output;
- $h_t \in R^h$ : vetor output da unidade LSTM;
- $\tilde{c}_t \in R^h$ : vetor de ativação da célula input;
- $c_t \in R^h$ : estado da célula;
- $W \in R^{h \times d}$ ,  $U \in R^{h \times h}$  e  $b \in R^h$ : matrizes de peso e vetor de parâmetros de bias, que precisam ser aprendidos durante o treinamento da rede neural;

Nestas equações, os subscritos  $d$  e  $h$  se referem ao número de variáveis de *input* e número de unidades na camada intermediária da rede, respectivamente. As funções de ativação, funções pré-definidas que tem como objetivo suavizar a transição de dados, que ocorre conforme os valores de *input* são modificados, são representadas por  $\sigma_g$  para a função sigmoide e  $\sigma_h$  para a função tangente hiperbólica.

### 2.2.3 LSTM bidirecional

Uma rede neural LSTM bidirecional (BI-LSTM) <sup>[22]</sup>, cujo fluxo de dados é representado na figura 2, pode ser sintetizada como a conjunção de duas redes neurais LSTM independentes, treinadas com os mesmos dados sequenciais, porém em sentidos opostos (do passado para o futuro e do futuro para o passado).

Seu diferencial em relação à rede LSTM unidirecional é que, na rede BI-LSTM, durante o processo de treinamento, os estados ocultos de ambas as redes são combinados, tornando assim possível a preservação da informação sobre o passado e sobre o futuro, gerando uma maior captação do contexto inerente aos dados.

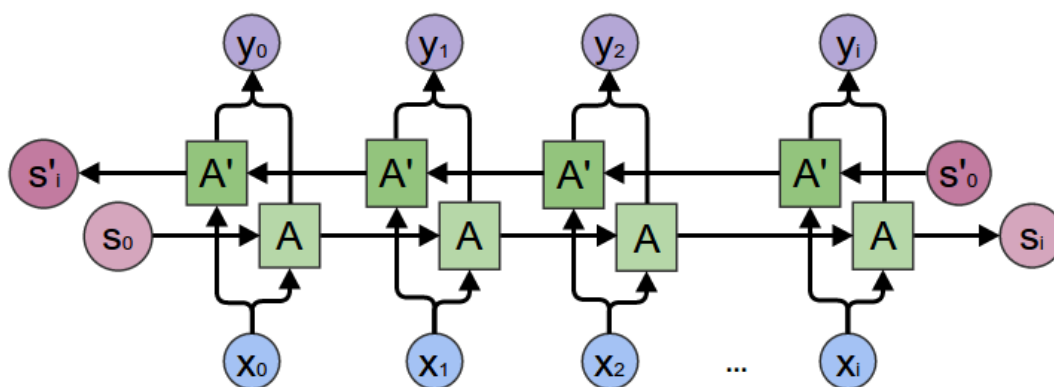


Figura 2 estrutura da rede LSTM bidirecional <sup>[23]</sup>

Esta modificação da arquitetura original costuma ser mais utilizada em tarefas de processamento de linguagem natural, geralmente com desempenho melhor que o da rede LSTM comum. No presente trabalho foi feita também uma investigação sobre o desempenho da rede LSTM bidirecional, na tarefa de previsão dos níveis de poluentes atmosféricos.

## 2.3 Gated Recurrent Unity

### 2.3.1 O funcionamento da rede GRU

Introduzida em 2014 por Cho *et al* <sup>[24]</sup>, a *Gated Recurrent Unity* (GRU) é mais uma das evoluções da rede neural do tipo recorrente. Assim como a LSTM, a GRU propõe uma maneira de diminuir o problema de dissipação do gradiente, que ocorre

quando os gradientes da rede neural param de ser atualizados, levando a um baixo desempenho do modelo.

Semelhantemente a LSTM, as unidades da GRU são baseadas em sistemas de portões de tráfego de informações. Estes portões, de atualização e redefinição, propiciam o descarte de informações irrelevantes à medida que o treinamento do modelo evolui.

Os dois portões são na prática, vetores que contém valores entre 0 e 1, que vão ser multiplicados pelos dados de entrada e/ou estado oculto. Um valor 0 nestes vetores indica que os dados devem ser descartados. Um valor 1, por sua vez, indica que os dados devem ser mantidos.

Como é mostrado na figura 3, ao contrário das unidades LSTM, que possuem um estado da célula (que carrega informações de longo prazo) e um estado oculto (que carrega informações de curto prazo), as unidades GRU, representadas na figura 4, possuem apenas um estado, que mantém as dependências de curto e longo prazo ao mesmo tempo.

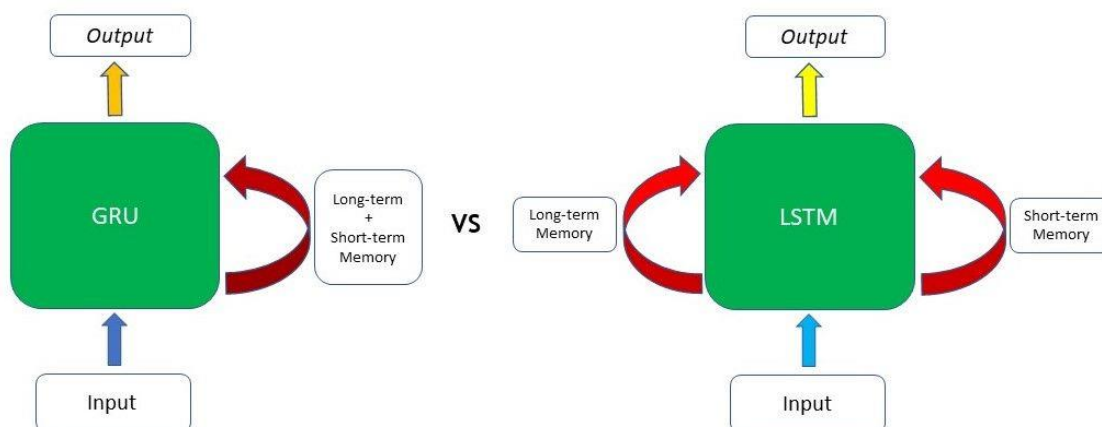


Figura 3 comparação entre GRU e LSTM <sup>[25]</sup>

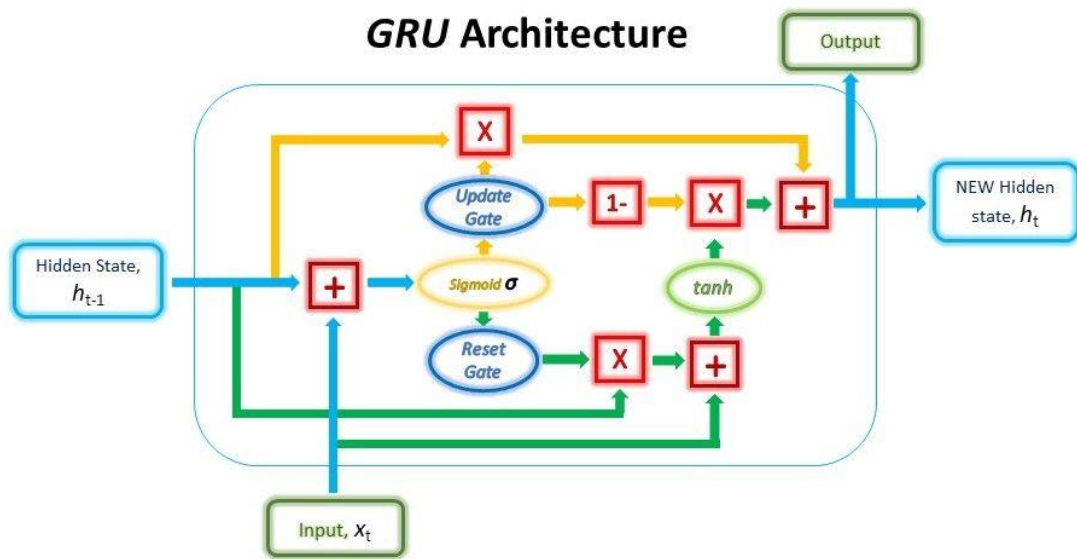


Figura 4 estrutura da unidade GRU <sup>[25]</sup>

A seguir, são explicados com maiores detalhes o funcionamento dos dois portões e como é feita a combinação de seus resultados, para gerar o *output* da rede.

### 2.3.2 Portão de redefinição

Na etapa de redefinição, o estado oculto da etapa anterior e a entrada atual são multiplicados por seus pesos e somados, antes de serem multiplicados pela função sigmóide. A função sigmóide vai colocar os valores no range de 0 a 1, e nas etapas subsequentes as informações serão descartadas de acordo com estes valores.

$$gate_{reset} = \sigma(W_{input_{reset}} \cdot x_t + W_{input_{reset}} \cdot h_{t-1}) \quad (16)$$

Em seguida, o estado oculto anterior é multiplicado por uma matriz de pesos treinável, passando então por uma multiplicação, elemento a elemento, com o vetor de redefinição  $gate_{reset}$ . Esta operação define que informação será passada adiante nas etapas seguintes, junto aos novos *inputs*.

Paralelamente, a entrada atual é multiplicada por uma matriz de pesos treinável e somada com o produto do vetor de redefinição e do estado oculto anterior. Por fim r,

que representa o portão de redefinição, é obtido ao aplicar-se a função tangente hiperbólica ao resultado desta etapa:

$$r = \tanh(\text{gate}_{reset} \odot (W_{h_1} \cdot h_{t-1}) + W_{x_1} \cdot x_t) \quad (17)$$

### 2.3.3 Portão de atualização

A fórmula utilizada para criar o vetor de atualização é a mesma que a do vetor de redefinição

$$\text{gate}_{update} = \sigma(W_{input_{update}} \cdot x_t + W_{hidden_{update}} \cdot h_{t-1}) \quad (18)$$

Porém, os pesos multiplicados pela entrada e pelo estado oculto são diferentes para cada portão. O vetor de atualização é submetido à multiplicação elemento a elemento, com o estado anterior, para obter  $u$ , que é utilizada na saída final:

$$u = \text{gate}_{update} \odot h_{t-1} \quad (19)$$

Este vetor é utilizado para determinar a quantidade de informações passadas, armazenadas no estado oculto anterior, que serão retidas para o futuro.

### 2.3.4 Combinação de resultados

Nesta última etapa, o novo estado oculto é finalmente obtido através da equação:

$$h_t = r \odot (1 - \text{gate}_{update}) + u \quad (20)$$

Após esta etapa, o novo estado oculto é utilizado como saída nesse intervalo de tempo, ao passar por uma camada de ativação linear.

Ao contrário das RNN tradicionais, onde todo o conteúdo do estado oculto é substituído a cada etapa, nas GRU uma parte deste é mantida, e é adicionado novo

conteúdo a ele. Com isto, os erros dos gradientes são retro propagados sem desaparecer ou explodir muito rapidamente.

## 3 Metodologia

### 3.1 Método de preenchimento de valores ausentes

Supondo que as estações mais próximas têm uma maior correlação em seus dados, o método para preencher os valores ausentes é constituído das seguintes etapas:

1. Obtenção da distância a estação desejada até as três estações mais próximas, utilizando a fórmula de Haversine<sup>[26]</sup>.

$$d = R \times c \quad (21)$$

Onde  $d$  é a distância entre as duas estações em quilômetros,  $R$  é o raio aproximado da Terra ( $\sim 6373$  km) e  $c$  é obtido através das relações:

$$a = \text{sen}(d_{lat}/2)^2 + \cos(lat1) \times \cos(lat2) \times \text{sen}(d_{lon}/2)^2 \quad (22)$$

$$c = 2 \times \arctan\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right) \quad (23)$$

Em que:

$$d_{lon} = lon2 - lon1 \quad (24)$$

$$d_{lat} = lat2 - lat1 \quad (25)$$

Onde  $lat1$  e  $lon1$  são as coordenadas geográficas UTM do ponto 1 em radianos e  $lat2$  e  $lon2$  são as coordenadas do ponto 2 em radianos.

2. Para cada variável foi feita uma média ponderada, utilizando os valores das 3 estações mais próximas para a variável, no tempo desejado e as suas respectivas distâncias, como na fórmula abaixo

$$x_A = \frac{x_B(1/d_{AB}) + x_C(1/d_{AC}) + x_D(1/d_{AD})}{(1/d_{AB}) + (1/d_{AC}) + (1/d_{AD})} \quad (26)$$

onde  $x$  representa o valor da variável e  $d$  a distância entre as estações

### 3.2 Escalonamento dos dados

A normalização e a padronização dos dados são técnicas aplicadas com frequência, na fase de preparação dos dados para o treinamento dos modelos. Em ambas as técnicas os dados são colocados numa mesma escala, geralmente com valores variando no range (0,1).

Este processo é importante, pois os dados trabalhados no problema possuem escalas diferentes, e caso não seja feita uma normalização, o modelo poderá dar um peso maior às variáveis de maior *range*, distorcendo seus resultados no treinamento.

Embora não seja uma etapa obrigatória, é de comum senso que a padronização dos dados pode influenciar positivamente nos resultados fornecidos por redes neurais, como pode ser comprovado em Shanker *et al.* [27]

Na padronização os dados são transformados de forma que a média dos registros seja zero e o desvio padrão seja 1. Este método é mais adequado para o escalonamento de variáveis cuja distribuição segue a curva gaussiana.

Uma vez que as variáveis que compõem o conjunto de dados estudado não possuem uma distribuição normal, para escalonar os dados, foi utilizado a normalização com mínimos e máximos, que coloca os dados no intervalo entre 0 e 1, sem causar nenhuma distorção nos mesmos. A fórmula da normalização por mínimo-máximo é dada por:

$$X_p = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (27)$$

### 3.3 Reframe do problema

Em etapas anteriores deste trabalho, o modelo foi montado para prever os níveis dos poluentes atmosféricos para até 8 horas no futuro. Com o intuito de aumentar a

utilidade do modelo em casos práticos, se fez necessária então uma reformulação do problema.

Desta forma, na versão mais atual do modelo, foi seguida a ideia vista em Navares *et al.* [28] e em Hossain *et al.* [29]. Em ambos os trabalhos foram utilizadas redes neurais do tipo recorrente, para prever os níveis médios diários de cada poluente atmosférico, em Madri (no caso do primeiro trabalho) e na zona portuária de Bangladesh (no caso do segundo trabalho).

Entretanto, uma vez que o objetivo do modelo foi prever a pior situação possível para o dia seguinte, foi feita a opção de prevêr a máxima diária para cada poluente, ao invés da média, como visto nos dois trabalhos.

### 3.4 Número de neurônios iniciais

Para que fosse obtida a estrutura de rede ideal para o modelo (número de camadas, número de neurônios em cada camada, tamanho do *batch* e número de épocas de treinamento), foram realizados testes preliminares. Para realizar estes experimentos, a metodologia utilizada foi a de variar os parâmetros estudados, enquanto os demais eram mantidos constantes.

Para uma estimativa inicial do número de neurônios de cada camada, nos primeiros testes, foi utilizada a fórmula presente no trabalho de Li *et al* [30]:

$$N_h = \frac{\sqrt{1 + 8N_i} - 1}{2} \quad (28)$$

Onde  $N_h$  é o número de neurônios da camada intermediária,  $N_i$  é o número de variáveis de input no conjunto de dados.

### 3.5 Tamanho do input de dados (LAG)

No presente trabalho, para a previsão das variáveis-alvo no tempo  $t$ , os *inputs* são o conjunto destas variáveis nos tempos  $t-8$  a  $t-1$ . Isto é, os últimos sete valores para

cada série que compõem a série temporal multivariada são vetorizados e alimentados ao modelo, para que sejam feitas previsões para as mesmas variáveis no tempo  $t$ .

### 3.6 Múltiplo input e múltiplo output

Devido à natureza do conjunto de dados, que possui múltiplas variáveis, relacionadas à meteorologia e a qualidade de ar local, a abordagem mais simples para o problema apresentado, envolveria a construção de um modelo para a previsão dos níveis de cada poluente separadamente.

No entanto, os pacotes de programação mais modernos para as redes neurais recorrentes, são capazes não só de receber múltiplas variáveis como *input*, como também de fornecer múltiplos *outputs*.

Desta forma, ao invés da construção de vários modelos, após o devido processamento de dados, utilizando a abordagem de *multiple input- multiple output*, como visto em Petersen *et al.* <sup>[31]</sup> foi possível realizar previsões para todas as variáveis de *input* em um mesmo modelo.

Depois de cumpridas as demais etapas, o conjunto de dados de *input* consistia de 12 variáveis. O *output* por sua vez também terá 12 variáveis. Uma vez que, os dados utilizados no problema foram agrupados por dia (máxima), o modelo recebe de input os últimos 7 dias e retorna a provisão para a máxima do dia seguinte.

Como temos 7 dias de input e 12 variáveis, uma das etapas de pré-processamento dos dados consistiu em formar o conjunto de input, onde cada registro terá 84 variáveis (os valores para os 7 últimos dias para cada uma das 12 variáveis). O output do modelo será então um vetor contendo as 12 previsões para cada uma das variáveis (máximas para o dia seguinte).

A ideia é que estas previsões sejam refeitas a cada hora, com novos dados horários e combinações, de forma que o modelo sempre terá previsões alinhadas à realidade ao longo do dia.

### 3.7 Validação cruzada

A validação cruzada é uma metodologia para avaliação de modelos de aprendizado de máquina, que é tida como mais confiável que a divisão simples dos dados em conjunto de treinamento e conjunto de teste. Existem diversas estratégias para implementá-la, uma delas é a validação cruzada do tipo k-fold.

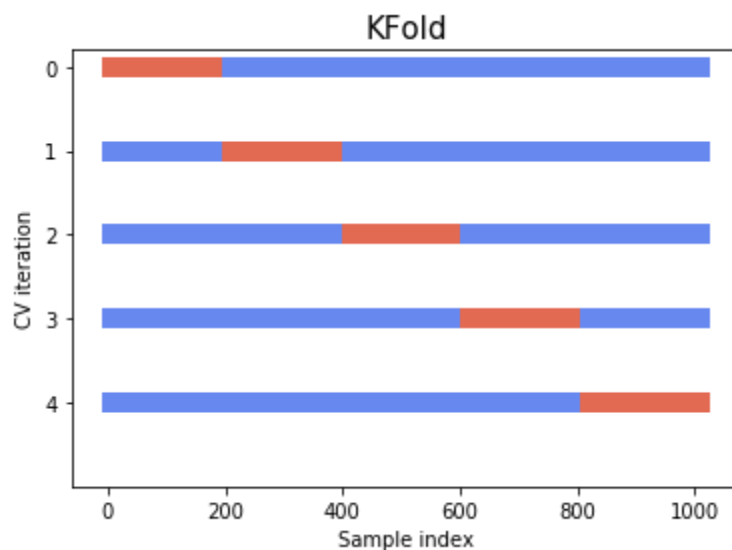


Figura 5 validação cruzada em dados não-sequenciais <sup>[32]</sup>

Neste método, representado na figura 5, o conjunto de treinamento é dividido em k partições. Durante cada iteração da validação cruzada, uma partição é utilizada para a validação e as demais são utilizadas para o treinamento. Isto permite uma melhor utilização dos dados. Com este método, também é evitado o viés do modelo, adquirido por algum padrão que possa estar presente em alguma das partições. O erro final do treinamento é a média dos erros de cada partição.

Todavia, para séries temporais a técnica k-fold não é aplicável, devido à natureza dos dados. Não é possível associar registros aleatoriamente para os conjuntos de treino e de teste porque não faria sentido utilizar valores do futuro para prever valores passados. Há uma dependência temporal entre as observações, que deverá ser preservada durante o teste.

Nos trabalhos de Cerqueira *et al.* [33] e Schnaubelt [34], dentre outras técnicas, foi apresentada uma solução para a utilização de validação cruzada com dados sequenciais, nomeada de *time series split*. A ideia por trás desta metodologia, representada na figura 6, é dividir o conjunto de treinamento em dois blocos em cada iteração, mantendo o bloco de treinamento sempre a frente, e aumentando a quantidade de dados utilizada a cada iteração, conforme mostra a figura:

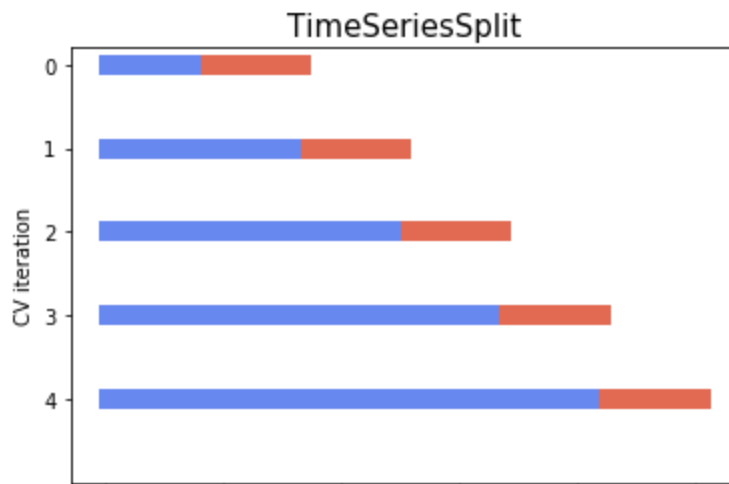
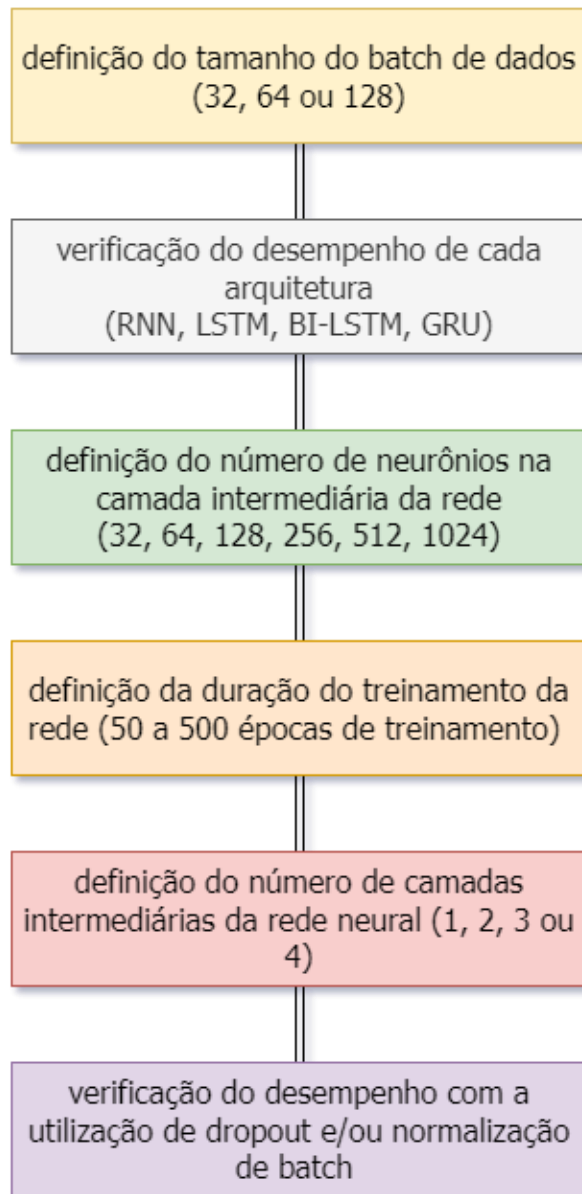


Figura 6 validação cruzada em séries temporais [32]

No presente trabalho, os modelos foram treinados com a validação cruzada de  $k=5$ , utilizando a técnica *time series split*.

### 3.8 Testes preliminares

Nesta etapa, buscou-se delimitar o melhor conjunto de hiperparâmetros para o modelo. Para todos os testes foi utilizada a técnica de validação cruzada com  $k=5$ . Para todas as redes o otimizador (algoritmo utilizado na otimização/mudança dos atributos da rede neural) utilizado foi o ADAM [35], com taxa de aprendizagem mantida fixa em 0.0005. O objetivo do modelo é a previsão dos níveis máximos no dia seguinte, para cada poluente e variável meteorológica. Os testes realizados estão resumidos na figura 7:



*Figura 7 testes realizados para definir melhor conjunto de hiperparâmetros*

Na primeira etapa dos testes, o objetivo foi encontrar um tamanho ótimo para o *batch* de dados (número de registros processados, antes que os pesos sejam atualizados), utilizado treinamento da rede neural recorrente. Para isto, foram mantidos fixos os demais hiperparâmetros e foram testados três tamanhos para o *batch* de dados: 32, 64 e 128.

Na etapa seguinte, mantendo-se fixos os demais hiperparâmetros, foram testadas diferentes arquiteturas para a rede neural recorrente (RNN, GRU, LSTM, BI-LSTM), com o objetivo de determinar os pesos de cada rede neural no modelo *ensemble*.

Na próxima etapa, foram mantidos fixos os demais hiperparâmetros e variou-se o número de neurônios na camada intermediária (de 32 a 1024), com o intuito de encontrar uma quantidade ótima para o número de neurônios na camada intermediária.

Na quarta etapa dos testes, mantidos fixos os demais hiperparâmetros, verificou-se o desempenho do modelo com diferentes durações de treinamento (de 50 a 500 épocas).

O próximo teste buscou determinar, mantendo-se os demais hiperparâmetros fixos, qual a quantidade ótima de camadas intermediárias na rede neural. Para isto foi estudado o desempenho de modelos com diferentes números de camadas (1 a 4).

Na última etapa, foram testadas as técnicas de *dropout*<sup>[36]</sup>, com diferentes percentuais (10 a 20%) e normalização de *batch*<sup>[37]</sup>, com o intuito de verificar se ambos métodos podem aprimorar o desempenho do modelo.

### 3.9 Ensemble de modelos

Devido a natureza estocástica dos algoritmos de aprendizagem das redes neurais, cada vez que um modelo é treinado, uma versão diferente da função que mapeia os *outputs* é aprendida, resultando em um desempenho diferente para os mesmos conjuntos de treino e teste.

Um método conhecido para reduzir esta variância, em modelos baseados em redes neurais, é treinar múltiplos modelos e combinar suas previsões. Esta combinação de previsões adiciona um viés que contrapõe a variância encontrada nas redes. Esta técnica, a qual se dá o nome de *ensemble*, reduz a variância das previsões, fornecendo melhores resultados que os de um modelo simples.

O *ensemble* de modelos costuma ser a maneira padrão aplicada a modelos de aprendizado de máquina, para garantir que melhores e mais estáveis previsões sejam feitas. Um exemplo notável é a rede convolucional AlexNet de 2012, na qual Alex

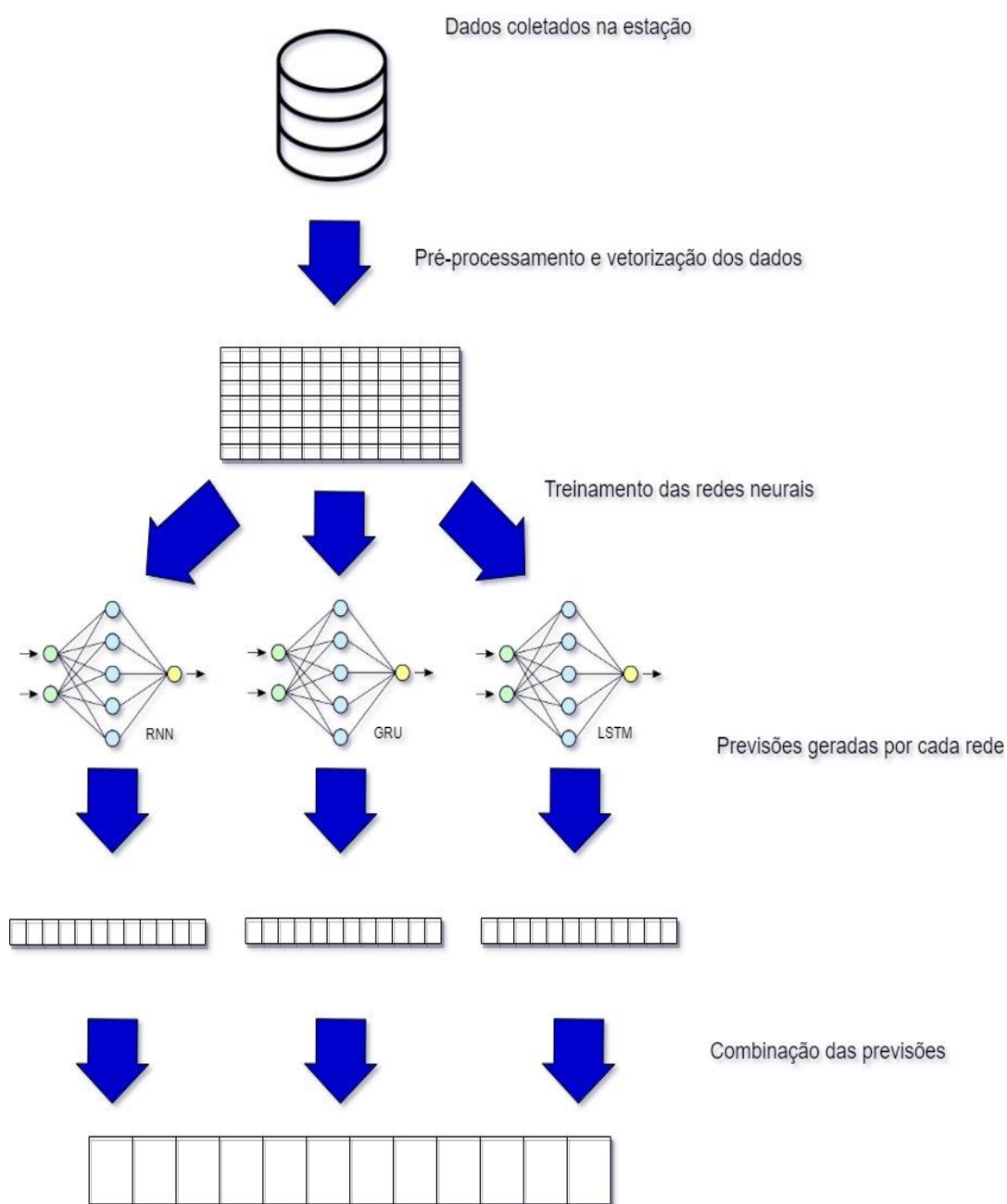
Krzhevsky *et al* <sup>[38]</sup> utilizou as médias dos resultados fornecidos pelos melhores modelos CNN à época e alcançou resultados equiparáveis ao estado-da-arte.

A maneira mais simples de combinar as previsões é calcular a média das previsões de cada modelo que compõe o *ensemble*. Isto pode ser levemente melhorado com a atribuição de pesos às previsões de cada modelo, conforme sugere Bishop <sup>[39]</sup>.

No presente trabalho, o modelo final se trata de um *ensemble* composto pelas redes neurais GRU, RNN e LSTM. Na primeira etapa da montagem do modelo, foram mapeados os desempenhos da cada rede neural separadamente. Na segunda etapa, de acordo com os resultados da etapa anterior, foram atribuídos pesos de 1 a 3 para cada rede, sendo 3 o peso atribuído ao melhor modelo e 1 ao pior modelo. Na etapa seguinte, os três modelos foram treinados novamente e seus resultados foram utilizados para calcular as médias ponderadas, fornecendo assim o resultado final.

$$M_p = \frac{(N_1 p_1) + (N_2 p_2) + (N_3 p_3)}{p_1 + p_2 + p_3} \quad (29)$$

A figura 8 mostra um esquema do ensemble dos resultados de todas as redes.



*Figura 8 esquema do ensemble dos resultados das redes neurais*

## 4 Estudo de caso: qualidade do ar no município do Rio

### 4.1 Contextualização

Em diversas cidades pelo mundo já existem projetos, potencialmente replicáveis localmente, que foram bem sucedidos ao utilizarem os dados de qualidade do ar. Na cidade de Chicago, o *Array of Things*<sup>[40]</sup> pretende gerar uma rede de sensores que vai disponibilizar, em tempo real, dentre outras, informações sobre a qualidade atmosférica. Em Pequim, o *IBM Green Horizons*<sup>[41]</sup>, tornou possível a identificação das fontes e previsão dos níveis de poluição atmosférica. No Reino Unido, o *Earthsense*<sup>[42]</sup> está estudando quais são os impactos na qualidade do ar, ao diminuírem-se as emissões de poluentes por parte dos veículos.

Um ponto a ser destacado, é a maneira pela qual alguns destes projetos coletam os dados. O projeto *Array of Things*, por exemplo, utiliza sensores estacionários acoplados a postes de iluminação pública para registrar as alterações atmosféricas. Outras iniciativas utilizam sensores móveis acoplados nos carros do *Google Street View*<sup>[43]</sup> ou em bicicletas compartilhadas<sup>[44]</sup> para reunir informações sobre a qualidade atmosférica pela cidade. Outra abordagem promissora, testada por pesquisadores da *Imperial College London*, é utilizar os sensores dos *smartphones* para identificar quais partes da cidade são mais poluídas<sup>[45]</sup>.

Todas estas metodologias podem oferecer uma alternativa viável à maneira como os dados são coletados atualmente, por meio de estações coletoras estáticas, que por serem mais caras, tendem a cobrir uma área menor do local de estudo. Uma configuração que já está sendo testada, é a utilização de sensores estáticos, como os encontrados nas estações do programa municipal local (MonitorAr Rio), para calibrar os resultados de sensores móveis, comumente utilizados em dispositivos wireless.

De posse destes dados sobre a qualidade do ar, a utilização mais comum parece ser a criação de sistemas de alerta<sup>[46]</sup> ou de previsão da concentração de poluentes<sup>[47]</sup>. Com estas informações, é possível elaborar, por exemplo, um modelo que gere avisos em tempo real, sobre eventos potencialmente prejudiciais à saúde, como o que é pretendido no presente trabalho.

## 4.2 MonitorAr-Rio

A prefeitura do Rio de Janeiro possui, desde 2017, um banco de dados públicos, conhecido como a Data.Rio. Mantido pelo Instituto Municipal de Urbanismo Pereira Passos <sup>[48]</sup>, o repositório possui conjuntos de dados relacionados à urbanização, economia, saúde e meio ambiente locais. Um destes conjuntos, possui o conjunto de dados de dados horários de monitoramento da qualidade do ar, que pertence ao projeto MonitorAr- Rio.

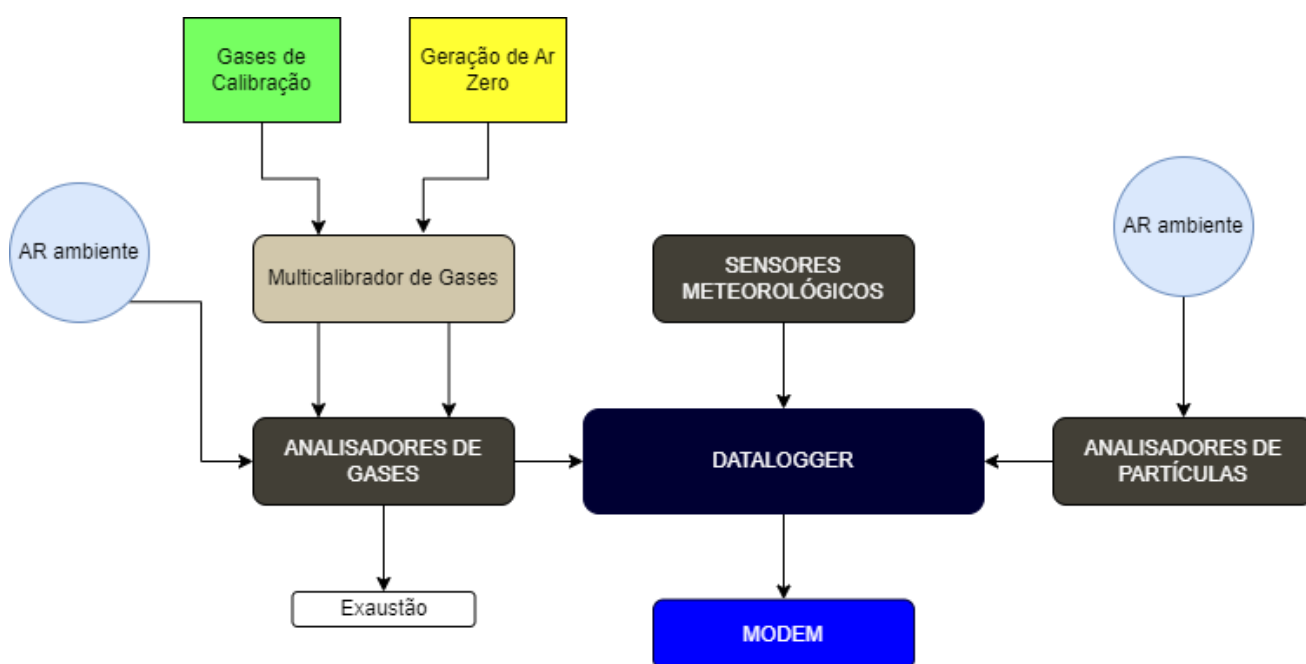
O projeto MonitorAr- Rio remonta originalmente ao ano de 1996. Neste ano, a Universidade Federal do Rio de Janeiro se tornou responsável pelo monitoramento das condições atmosféricas do município. Em 2000, a UFRJ iniciou um estudo sobre das concentrações de ozônio no Rio, utilizando amostradores passivos <sup>[49]</sup>. Por conta deste projeto, foram instaladas as estações de São Cristóvão, Tijuca, Centro e Copacabana. Em 2008, em convênio com a Petrobrás, a prefeitura adicionou mais quatro estações restantes às originais, em Campo Grande, Irajá, Bangu e Pedra de Guaratiba. A localização geográfica de cada estação está representada na figura 9.



Figura 9 localização das estações do Monitorar-Rio <sup>[50]</sup>

Estas estações, em conjunto com uma estação móvel, registram a meteorologia local e variações nas concentrações de cinco poluentes: ozônio, dióxido de enxofre, monóxido de carbono, material particulado e dióxido de nitrogênio.

A transmissão e o processamento dos dados da rede, representados na figura 10, são realizados por meio de telemetria. Estes dados são registrados com frequência amostral de dez minutos e então são enviados ao centro de supervisão MonitorAr- Rio. Todas as estações da rede são compostas por analisadores de partículas e gases, além dos sensores meteorológicos. Estes equipamentos operam de forma automática e geram dados continuamente. Os dados gerados são validados, a cada 15 minutos através da avaliação dos analisadores de poluentes atmosféricos, e o monitor é calibrado conforme a qualidade dos resultados obtidos.



*Figura 10 funcionamento da rede de captação de dados*

Existem atualmente três resoluções do Conselho Nacional do Meio Ambiente (CONAMA) <sup>[51]</sup>:

- A Resolução CONAMA Nº 018/1986, que dispõe sobre a criação Programa de Controle de Poluição do Ar por Veículos Automotores (PROCONVE);
- A Resolução CONAMA Nº 005/1989, que dispõe sobre o Programa Nacional de Controle da Poluição do Ar (PRONAR);
- A Resolução CONAMA Nº 491/2018, que estabelece os limites máximos de emissão de poluentes atmosféricos para fontes fixas;

A resolução de 2018 estabelece limites para concentrações dos principais poluentes, que devem ser coletados em quatro etapas separadas, produzindo três padrões intermediários e um padrão final, conforme consta na figura 11:

Poluente Atmosférico	Período de Referência	PI-1	PI-2	PI-3	PF	
		µg/m <sup>3</sup>	µg/m <sup>3</sup>	µg/m <sup>3</sup>	µg/m <sup>3</sup>	ppm
Material Particulado - MP <sub>10</sub>	24 horas	120	100	75	50	-
	Anual <sup>1</sup>	40	35	30	20	-
Material Particulado - MP <sub>2,5</sub>	24 horas	60	50	37	25	-
	Anual <sup>1</sup>	20	17	15	10	-
Dióxido de Enxofre - SO <sub>2</sub>	24 horas	125	50	30	20	-
	Anual <sup>1</sup>	40	30	20	-	-
Dióxido de Nitrogênio - NO <sub>2</sub>	1 hora <sup>2</sup>	260	240	220	200	-
	Anual <sup>1</sup>	60	50	45	40	-
Ozônio - O <sub>3</sub>	8 horas <sup>3</sup>	140	130	120	100	-
Fumaça	24 horas	120	100	75	50	-
	Anual <sup>1</sup>	40	35	30	20	-
Monóxido de Carbono - CO	8 horas <sup>3</sup>	-	-	-	-	9
Partículas Totais em Suspensão - PTS	24 horas	-	-	-	240	-
	Anual <sup>4</sup>	-	-	-	80	-
Chumbo - Pb <sup>5</sup>	Anual <sup>1</sup>	-	-	-	0,5	-
<sup>1</sup> - média aritmética anual						
<sup>2</sup> - média horária						
<sup>3</sup> - máxima média móvel obtida no dia						
<sup>4</sup> - média geométrica anual						
<sup>5</sup> - medido nas partículas totais em suspensão						

Figura 11 padrões de qualidade do ar do CONAMA <sup>[52]</sup>

Um dos objetivos do projeto MonitoAr é a certificação de que os limites estabelecidos nesta resolução, não sejam ultrapassados.

### 4.3 Variáveis meteorológicas e poluentes atmosféricos

#### 4.3.1 Temperatura

A temperatura afeta o movimento do ar, que por sua vez afeta a poluição atmosférica. Uma vez que a energia solar é absorvida pela superfície terrestre, o ar junto à superfície é mais quente que o ar da troposfera. O ar mais quente (e mais leve) da superfície sobe e o ar mais frio (e mais pesado) na troposfera desce. Este efeito é conhecido como convecção e move os poluentes das altitudes mais baixas para as mais altas.

Alguns poluentes, como o ozônio, são produzidos mais abundantemente em temperaturas mais quentes. As reações que criam o ozônio requerem luz solar. No verão e especialmente durante ondas de extremo calor, o ozônio alcança níveis perigosos.

Secas também podem ocorrer durante uma onda de calor e durante estas, incêndios florestais são mais comuns. Estes incêndios lançam monóxido de carbono e materiais particulados na atmosfera.

#### ***4.3.2 Pressão atmosférica***

A pressão atmosférica é a força exercida pela coluna de ar acima de nós. É resultante da colisão entre das moléculas presentes no gás atmosférico e de outros corpos nele inseridos. É diretamente proporcional à temperatura, com isto há uma tendência para menores pressões atmosféricas em locais de altitudes mais altas.

Baixas pressões estão associadas a condições úmidas e com ventos. Uma tempestade pode mover poluentes da atmosfera, levando-os para outras áreas. Altas pressões podem criar ar estagnado e concentrações de poluentes locais.

#### ***4.3.3 Velocidade do vento***

O vento é o movimento do ar que é causado pelas diferenças de pressão atmosférica. Quantidades de ar frio e quente se formam quando a luz solar atinge a terra, estas quantidades são desiguais devido aos fatores geográficos da superfície.

As diferenças de temperatura causam diferenças de pressão, o ar quente sobe e deixa para trás uma área de baixa pressão. Em áreas de baixas pressões, os gases no ar são menos densos. Devido ao princípio da difusão, gases se movem das áreas com alta pressão para as de baixa pressão. Quanto maior a diferença na pressão entre duas áreas, mais rapidamente os gases se movem, criando vento.

Áreas costeiras ou regiões com poucas obstruções geográficas tendem a ter climas com mais vento, e conseqüentemente, uma melhor qualidade do ar, uma vez que o vento carrega muito da poluição originada na área (prédios atrapalham). Desta forma, devido ao vento, a poluição originada em um local pode afetar também outras áreas.

Maiores velocidades do vento, geralmente se traduzem em uma maior dispersão dos poluentes, resultando em uma maior concentração de poluentes em áreas com menos ventos. Enquanto o solo esquenta durante o dia, o ar geralmente fica mais turbulento, causando a dispersão de poluentes no ar. Quando o ar se esfria a noite, condições mais estáveis ocorrem, causando menor dispersão de poluentes.

Entender a velocidade e direção do vento pode auxiliar na identificação da origem de poluentes e na projeção de tendências para pontos futuros, em uma determinada área.

#### **4.3.4 Chuva**

O processo de formação de chuva se inicia quando a água é aquecida e se transforma em vapor de água. Este vapor, por ser mais leve, sobe e forma nuvens carregadas de água. Quando estas nuvens atingem altitudes mais elevadas, a água se condensa novamente e se precipita.

Em dias chuvosos, os poluentes atmosféricos são eliminados da atmosfera, melhorando a qualidade do ar, num processo conhecido como deposição úmida. Neste processo natural, os materiais são removidos por hidrometeoros (gotas de chuva, cristais de gelo, neve etc.) e transferidos para a superfície terrestre.

#### **4.3.5 Umidade relativa do ar**

Umidade pode ser definida como, a quantidade de vapor de água dispersa pelo ar. A umidade tem influência direta sobre a temperatura, sensação térmica e sobre os períodos com chuva. A água em forma de vapor, presente no ar, determina a umidade relativa, que é a relação entre a quantidade de água que existe no ar e a quantidade máxima de água que poderia existir na mesma temperatura.

Baixa umidade do ar pode provocar problemas de saúde como irritações na garganta e sangramentos no nariz. Por outro lado, umidade em excesso pode originar problemas do trato respiratório como a rinite ou a asma.

Além disso, a umidade alta aumenta a taxa de químicos tóxicos no ar e amplifica o efeito negativo de poluentes atmosféricos como o *smog*. A umidade também influencia a deposição de material particulado no ar, pois quanto maior a umidade

relativa, maior será o tamanho do material particulado. Eventualmente, o material particulado pode ficar tão pesado para o ar, que se iniciará sua deposição, num fenômeno conhecido como deposição seca.

#### **4.3.6 Óxidos de nitrogênio**

Os óxidos de nitrogênio (NO<sub>x</sub>) são gases que contém nitrogênio e oxigênio em sua composição molecular. Dentre os gases deste grupo, os maiores poluentes atmosféricos são o dióxido de nitrogênio (NO<sub>2</sub>) e o óxido nítrico (NO).

Ambos gases são gerados por processos naturais como descargas elétricas e transformações microbianas. Todavia, atividades humanas como a queima de combustíveis fósseis, possuem a capacidade de elevar o nível de óxidos de nitrogênio na atmosfera artificialmente. Os NO<sub>x</sub> estão associados a doenças do trato respiratório, como edemas pulmonares.

Estes gases podem reagir com os radicais de hidroxila presentes no vapor d'água, causando assim chuva ácida, pela formação de ácido nítrico. Além disso, assim como ocorre com o ozônio, o NO<sub>x</sub> participa da formação de *smog* fotoquímico. Os gases NO e NO<sub>2</sub> contribuem ainda para a degradação da camada de ozônio, uma vez que podem reagir com moléculas de oxigênio e ozônio, presentes na estratosfera.

#### **4.3.7 Monóxido de carbono**

O monóxido de carbono (CO) é um gás incolor e sem odor, que pode ser prejudicial à saúde quando inalado em grandes quantidades. As maiores fontes deste poluente são as queimadas florestais e a queima de combustíveis fósseis.

Respirar um ar com alta concentração de monóxido de carbono reduz a quantidade de oxigênio que é transportada para a corrente sanguínea e a órgãos, como o cérebro e o coração. Em níveis muito altos, o monóxido de carbono pode causar tontura, confusão mental, desmaios e até óbitos.

Quando seus níveis na atmosfera estão altos, o CO pode ser especialmente preocupante para pessoas que possuem doenças cardíacas. Estas pessoas já podem possuir naturalmente baixa oxigenação no sangue, e o CO pode piorar essa condição, levando a um quadro de angina (dor no peito).

#### **4.3.8 Enxofre**

O dióxido de enxofre é um gás incolor, que provém primariamente da queima de combustíveis fósseis, como o diesel, o carvão e o petróleo. Este gás também pode ser produzido naturalmente por erupções vulcânicas.

Ao entrar em contato com o oxigênio atmosférico o enxofre pode reagir com a umidade do ar e formar ácido sulfúrico e retornar para a terra na forma de chuva ácida. O SO<sub>2</sub> pode também reagir com outros compostos atmosféricos e formar material particulado.

Além de contribuir para o aquecimento do planeta e formar a chuva ácida, o gás é altamente solúvel nas mucosas do trato aéreo, podendo provocar irritação e agravar problemas respiratórios e cardiovasculares.

#### **4.3.9 Material particulado**

A Agência de Proteção Ambiental dos Estados Unidos (EPA) classifica o material particulado (PM) como uma mistura de diversos materiais cujo diâmetro é menor que 10 micrômetros. As partículas de PM podem ser formadas por compostos químicos orgânicos, ácidos, metais e poeira. As principais fontes de emissão de materiais particulados são a poeira, a queima de combustíveis fósseis e a agricultura.

Os materiais particulados são divididos em duas subcategorias. O PM<sub>2,5</sub> é formado por partículas de até 2,5 micrômetros de diâmetro e geralmente é encontrado em fumaças e nevoeiros. O PM<sub>10</sub> possui partículas de 2,5 a 10 micrômetros de diâmetro e é encontrado em regiões industriais.

Ao lado do CO<sub>2</sub>, a fuligem, um material particulado proveniente da queima incompleta do diesel e dos incêndios, é uma das principais causas do aumento do aquecimento global. Dentre outros problemas ambientais, o PM também é apontado como causa da diminuição da frequência de chuvas e também da chuva ácida.

O material particulado é associado ao aparecimento de doenças como ataques cardíacos, arritmias cardíacas, asma, irritação das vias aéreas, tosses e dificuldades de respiração.

#### **4.3.10 Ozônio atmosférico**

O ozônio (O<sub>3</sub>) é um gás que pode ser encontrado naturalmente entre 15 km e 50 km de altitude (estratosfera), onde tem função de absorver a radiação solar e impedir a chegada dos raios ultravioletas emitidos pelo sol a superfície. No entanto, quando presente na camada inferior da atmosfera (troposfera), o ozônio é um poluente nocivo à saúde humana.

Seu ciclo de formação se desencadeia através da ação da luz solar sob os átomos de oxigênio, que são liberados pelo dióxido de nitrogênio, e se combinam ao oxigênio presente na atmosfera, formando o ozônio. O ciclo se completa quando óxido de nitrogênio se combina mais uma vez ao ozônio, formando novamente o dióxido de nitrogênio.

Por haver ação direta da luz solar na formação deste gás, se diz que o ozônio, quando encontrado na troposfera, é um oxidante fotoquímico. Esta categoria de poluentes, que tem sua origem ligada principalmente à queima de combustíveis fósseis e à agricultura, é a causa da chamada névoa fotoquímica ou *smog*, popularmente conhecida por diminuir a visibilidade dos motoristas em centros urbanos.

A exposição prolongada a esta névoa fotoquímica pode causar o aparecimento de doenças respiratórias e cardiovasculares, como a asma, a enfisema pulmonar e a arteriosclerose. Estes problemas se tornam ainda mais graves nos meses de verão e em climas quentes, uma vez que a luz solar é catalisadora da reação que gera o ozônio troposférico.

### **4.4 Arquitetura sugerida para o modelo em produção**

O esquema da figura 12 exibe o esboço da arquitetura do modelo em produção, hospedado na nuvem. Por uma questão de familiaridade com a plataforma, as ferramentas a seguir fazem parte da *Google Cloud*, todavia o esquema poderia ser posto em prática com quaisquer ferramentas análogas, de outros provedores de computação em nuvem. Adiante, são explicadas brevemente as ferramentas presentes no fluxograma.

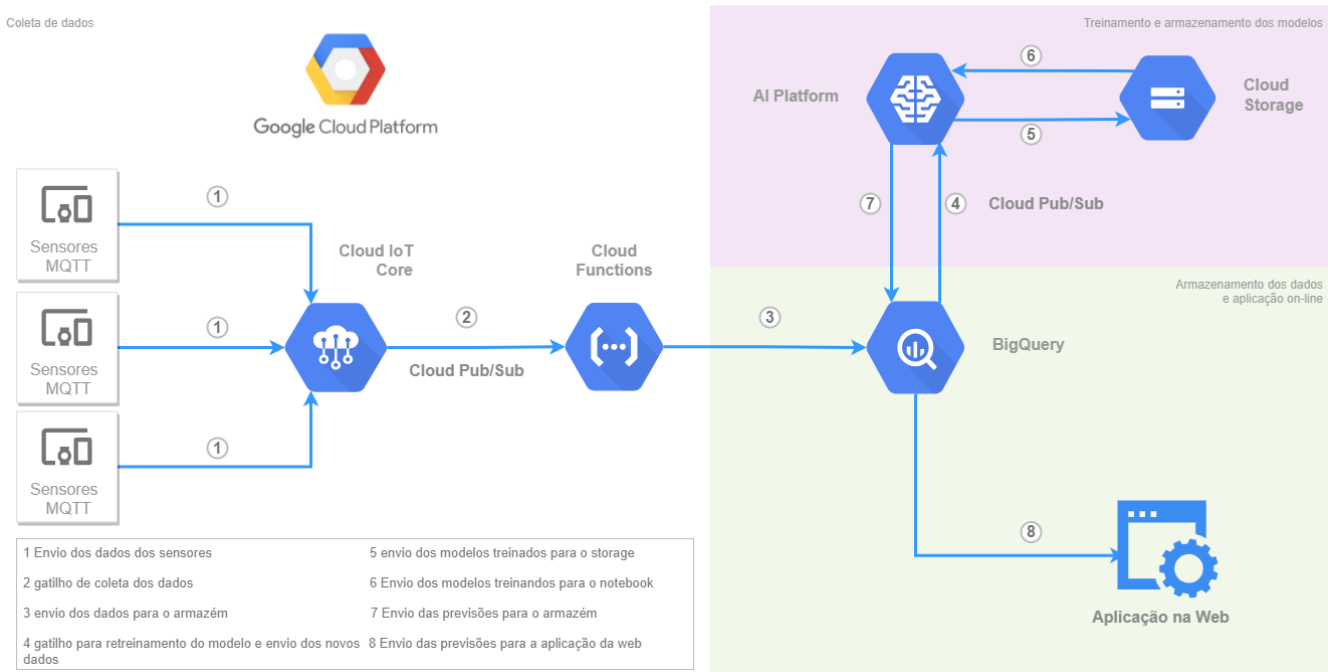


Figura 12 arquitetura sugerida para a aplicação em produção

O MQTT (*Message Queue Telemetry Transport*) é um protocolo de comunicação entre máquinas, especialmente utilizado em implementações de Internet das Coisas (IoT).

Este protocolo permite o envio de mensagens e comandos entre dispositivos que MQTT por meio de TCP/IP. Dentre as relatadas vantagens na utilização deste protocolo, empregado com sucesso na coleta de dados de qualidade do ar por Muladi *et al.* [53], estão a maior segurança e a melhor comunicação entre os dispositivos da rede.

O Cloud Iot Core é um serviço da Google, utilizado para conectar e administrar dispositivos de Internet das Coisas, como sensores telemétricos utilizados pelo projeto MonitorAr.

O Google Pub/Sub é um serviço de envio e recebimento de mensagens em tempo real. Com este serviço é possível, dentre outras tarefas, o encadeamento e agendamento das etapas constituintes de um *pipeline* de dados.

O Google Cloud Functions é um ambiente de execução sem servidor, utilizado para criar e conectar serviços em nuvem. Com este serviço, é possível escrever funções

de finalidade definida, atreladas a eventos emitidos pela infraestrutura em nuvem e pelos serviços. A função é executada quando determinado evento ocorre.

O Google BigQuery é um serviço de armazenamento de dados corporativo. Sua arquitetura sem servidor permite utilizar consultas SQL (*structured query language*) para responder a perguntas da organização sem a necessidade do gerenciamento de infraestrutura. O mecanismo de análise distribuída e escalonável do BigQuery permite a execução de consultas de terabytes em segundos.

A AI Platform é a plataforma da Google, direcionada ao desenvolvimento de modelos de aprendizado de máquina. Nesta plataforma é possível, dentre outras aplicações, utilizar ferramentas como notebooks Jupyter e diversas bibliotecas do Python como a TensorFlow e a Keras.

O Google Cloud Storage é um serviço para armazenamento de arquivos de diversos formatos, na nuvem do Google.

Como mostrado no esquema, os dados percorrem o seguinte fluxo:

- Os sensores presentes nas estações de coleta de dados de qualidade do ar enviam os dados para o provedor na nuvem, utilizando o protocolo MQTT.
- Na Google Cloud, os dados passam pela ferramenta Cloud IoT Core, que faz a conexão com os demais componentes da arquitetura.
- A cada hora um *job* é executado, para que os dados horários sejam enviados ao BigQuery e deste para a AI Platform, que atualizará a previsão fornecida pelo modelo de aprendizado de máquina, já treinados e salvos no Cloud Storage.
- Periodicamente, um *job* agendado é executado para que o modelo seja re-treinado, atualizando os pesos das redes neurais.
- As previsões fornecidas pelo modelo a cada hora, ficarão salvas no BigQuery, para que deste possam ser transferidas para uma aplicação on-line, como o atual site do MonitorAr.

## 5 Análise exploratória dos dados

### 5.1 Descrição dos dados

A montagem dos modelos, assim como a análise dos dados, foi feita com o emprego da linguagem Python. Na análise exploratória de dados foram utilizadas as bibliotecas Pandas, Matplotlib e Seaborn. As redes neurais foram implementadas com a biblioteca Keras. Os notebooks com as análises e os resultados das redes neurais foram montados com Google Colab. Estes notebooks, bem como as planilhas completas com os resultados de cada rede neural, podem ser encontrados no seguinte repositório do Github: <https://github.com/josefernandobr/notebooks-e-resultados>.

O conjunto de dados utilizado está atualmente disponível no repositório de dados da prefeitura (Datario). As informações foram coletadas no período de janeiro de 2012 a junho de 2020, em oito diferentes estações (Centro, Copacabana, São Cristóvão, Tijuca, Irajá, Bangu, Campo Grande e Pedra de Guaratiba). Os parâmetros monitorados (médias horárias) em cada estação são:

- Centro (CA): O<sub>3</sub>, CO, PM<sub>10</sub>
- Copacabana (AV): SO<sub>2</sub>, O<sub>3</sub>, CO, PM<sub>10</sub>
- São Cristóvão (SC): SO<sub>2</sub>, O<sub>3</sub>, CO, PM<sub>10</sub>
- Tijuca (SP): SO<sub>2</sub>, NO<sub>x</sub>, O<sub>3</sub>, CO, PM<sub>10</sub>
- Irajá (IR): SO<sub>2</sub>, NO<sub>x</sub>, O<sub>3</sub>, CO, HC, PM<sub>2.5</sub>, PM<sub>10</sub>
- Bangu (BG): SO<sub>2</sub>, NO<sub>x</sub>, O<sub>3</sub>, CO, HC, PM<sub>10</sub>
- Campo Grande (CG): SO<sub>2</sub>, NO<sub>x</sub>, O<sub>3</sub>, CO, HC, PM<sub>10</sub>
- Pedra de Guaratiba (PG): O<sub>3</sub>, PM<sub>10</sub>

Em que:

- CO: monóxido de carbono [ppm]
- HC: hidrocarbonetos [ppm]
- PM<sub>2.5</sub>: material particulado de até 2.5 micrômetros [ $\mu\text{g}/\text{m}^3$ ]
- PM<sub>10</sub>: material particulado de até 10 micrômetros [ $\mu\text{g}/\text{m}^3$ ]
- O<sub>3</sub>: ozônio [ $\mu\text{g}/\text{m}^3$ ]

- SO<sub>2</sub>: dióxido de enxofre [ug/m<sup>3</sup>]
- NO<sub>x</sub>: óxidos de nitrogênio [ug/m<sup>3</sup>]

O analisador de hidrocarbonetos (HC) realiza medidas dos gases hidrocarbonetos não-metano (HCNM), metano (CH<sub>4</sub>) e hidrocarbonetos totais (HCT). O analisador de óxidos de nitrogênio (NO<sub>x</sub>) realiza medidas de concentração dos gases monóxido de nitrogênio (NO), dióxido de nitrogênio (NO<sub>2</sub>) e óxidos de nitrogênio (NO + NO<sub>2</sub> = NO<sub>x</sub>).

Além dos poluentes, as estações de monitoramento da qualidade do ar registram também as variações horárias nas médias das seguintes variáveis meteorológicas:

- Vel\_Vento: velocidade do vento [m/s];
- Dir\_Vento: direção do vento [°];
- RS: radiação solar [w/m<sup>2</sup>];
- Chuva: precipitação pluviométrica [mm];
- UR: umidade relativa do ar [%];
- Temp: temperatura [°C];
- Pres: pressão atmosférica [mbar];

Por fim, o conjunto de dados apresenta também, a latitude, longitude e coordenadas UTM de cada estação, que não foram utilizadas neste momento. A tabela 1 resume as principais características do conjunto de dados utilizado:

**Tabela 1** atributos do conjunto de dados de monitoramento de poluição atmosférica

<b>Categoria do conjunto de dados</b>	Multivariado, sequencial, série temporal
<b>Número de registros</b>	616517
<b>Tipo de atributos</b>	Real
<b>Número de atributos</b>	22
<b>Tarefa associada</b>	Previsão
<b>Valores faltantes</b>	Sim

## 5.1 Resumo estatístico

Pela análise das tabelas 2 a 5, é notável que existem valores ausentes em grande parte das variáveis dos conjunto de dados. É evidente também que o range dos valores assumidos por cada coluna varia consideravelmente, o que pode demandar também uma

padronização destes dados, para que o modelo tenha um melhor desempenho, como foi visto anteriormente.

Pode ser observado também que a distribuição dos dados em diferentes estações é bastante similar, abrindo a possibilidade de se utilizarem os dados de estações adjacentes no preenchimento de variáveis ausentes. As variáveis ausentes em cada estação foram omitidas das tabelas a seguir. Por razões explicitadas mais adiante, nas etapas seguintes foram utilizados apenas os dados das quatro estações, cujas estatísticas são exibidas nas tabelas 2 a 5.

**Tabela 2** Resumo estatístico dos dados da estação do Centro

	Temp	CO	Chuva	Pres	Vel_Vento	SO <sub>2</sub>	PM <sub>10</sub>	O <sub>3</sub>	UR
<b>contagem</b>	62045,0	77084,0	80790,0	80290,0	79951,0	35688,0	76444,0	76827,0	61888,0
<b>média</b>	25,1	0,4	0,1	1014,9	1,0	6,0	33,3	20,9	69,1
<b>std</b>	4,4	0,3	1,0	4,8	0,5	7,4	22,2	21,8	16,1
<b>mínimo</b>	13,2	0,0	0,0	999,0	0,0	0,0	0,0	0,0	0,0
<b>25%</b>	21,9	0,2	0,0	1011,5	0,6	1,9	20,0	4,4	58,3
<b>50%</b>	24,6	0,4	0,0	1014,5	0,8	3,9	30,0	14,2	71,5
<b>75%</b>	27,8	0,5	0,0	1018,1	1,3	7,3	42,0	30,6	81,6
<b>máximo</b>	44,3	4,4	65,6	1032,5	4,9	108,5	994,0	304,6	100,0

**Tabela 3** Resumo estatístico dos dados da estação de Copacabana

	Temp	CO	Chuva	Pres	Vel_Vento	SO <sub>2</sub>	PM <sub>10</sub>	O <sub>3</sub>	UR
<b>contagem</b>	67044,0	77426,0	72308,0	72359,0	71804,0	70916,0	78281,0	78200,0	66768,0
<b>média</b>	25,3	0,2	0,1	1015,3	0,7	3,5	54,6	19,9	73,3
<b>std</b>	4,1	0,2	1,1	4,8	0,3	4,9	21,9	16,5	14,3
<b>mínimo</b>	0,0	0,0	0,0	999,7	0,0	0,0	0,0	0,0	19,2
<b>25%</b>	22,4	0,0	0,0	1011,9	0,5	0,7	39,6	7,0	64,2
<b>50%</b>	24,8	0,1	0,0	1014,9	0,6	1,9	50,8	16,6	74,9
<b>75%</b>	27,6	0,2	0,0	1018,5	0,9	4,3	65,6	28,9	83,9
<b>máximo</b>	42,6	3,1	56,2	1032,6	12,4	124,3	296,5	228,6	100,0

**Tabela 4** Resumo estatístico dos dados da estação de São Cristóvão

	Temp	CO	Chuva	Pres	Vel_Vento	SO <sub>2</sub>	PM <sub>10</sub>	O <sub>3</sub>	UR
<b>contagem</b>	74269,0	77622,0	80811,0	81198,0	80712,0	73961,0	76940,0	77542,0	70389,0
<b>média</b>	26,8	0,2	0,1	1015,0	0,9	8,5	34,1	23,9	73,2
<b>std</b>	4,8	0,3	0,9	4,8	0,6	10,2	22,2	27,0	17,1
<b>mínimo</b>	0,0	0,0	0,0	999,0	0,0	0,0	0,0	0,0	0,0
<b>25%</b>	23,4	0,1	0,0	1011,6	0,5	2,4	19,0	2,9	61,7
<b>50%</b>	26,3	0,1	0,0	1014,6	0,8	5,5	29,0	14,4	74,7
<b>75%</b>	29,7	0,3	0,0	1018,2	1,2	10,7	44,0	36,9	86,4
<b>máximo</b>	49,1	4,7	55,8	1032,6	4,5	276,6	338,0	284,4	100,0

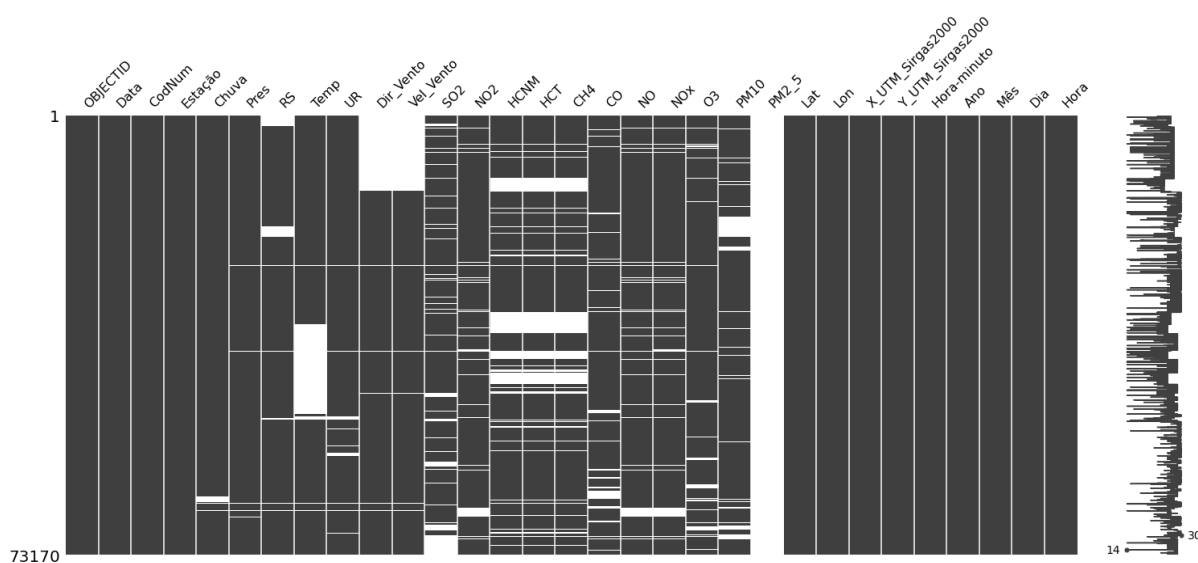
**Tabela 5** Resumo estatístico dos dados da estação da Tijuca

	Temp	NOx	CO	Chuva	Pres	Vel_Vento	SO <sub>2</sub>	PM <sub>10</sub>	O <sub>3</sub>	UR	NO	NO <sub>2</sub>
<b>count</b>	79741,0	48553,0	79121,0	79234,0	80195,0	66838,0	72115,0	79506,0	77057,0	80588,0	48550,0	48550,0
<b>mean</b>	24,4	57,7	0,4	0,1	1015,2	0,7	3,8	32,9	28,9	66,5	35,5	42,3
<b>std</b>	5,6	36,2	0,3	3,6	5,4	1,2	6,0	19,0	25,4	19,5	26,6	24,2
<b>min</b>	11,3	0,0	0,0	0,0	997,5	0,0	0,0	0,0	0,0	0,0	0,0	0,0
<b>25%</b>	20,3	32,1	0,2	0,0	1011,3	0,3	0,8	20,4	10,3	53,4	14,6	24,2
<b>50%</b>	23,3	51,2	0,4	0,0	1014,9	0,5	2,0	29,1	23,2	68,3	30,1	38,7
<b>75%</b>	27,5	74,9	0,5	0,0	1018,9	0,9	4,4	40,7	39,9	81,5	51,0	55,4
<b>max</b>	46,2	418,5	12,1	426,6	1036,5	25,5	159,8	310,3	295,5	100,0	241,8	235,9

## 5.2 Estado do preenchimento do conjunto de dados

Nas figuras à seguir estão representados os estados de preenchimentos dos subconjuntos de dados, referentes a cada estação. A primeira parte de cada figura consiste num mapa de valores ausentes, em que a cor branca representa os registros que não estão presentes. A barra ao lado desta representação mostra o número de colunas com valores ausentes em cada registro.

Como fica evidente nas figuras 13 a 20, por razão da ausência da coleta de dados para determinadas estações, ou de dados espúrios para outras variáveis, se faz necessária à adoção de uma estratégia para o preenchimento dos valores ausentes. A metodologia adotada para o tratamento dos valores ausentes foi explicada no tópico 3.1 deste trabalho.



*Figura 13 grau de preenchimento dos dados da estação de Bangu*

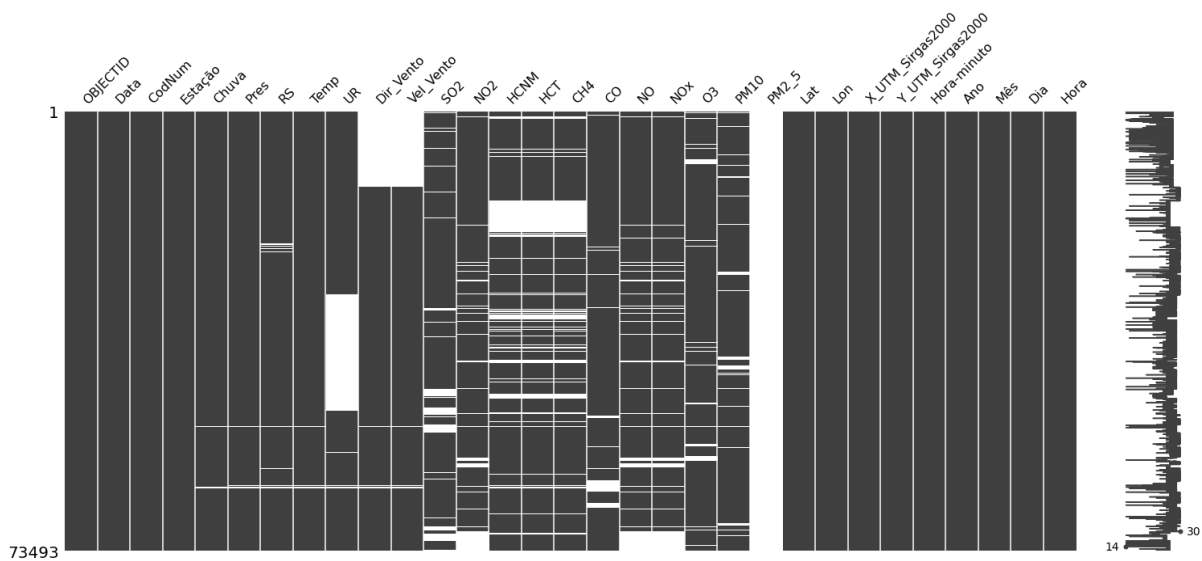


Figura 15 grau de preenchimento da estação de Campo Grande

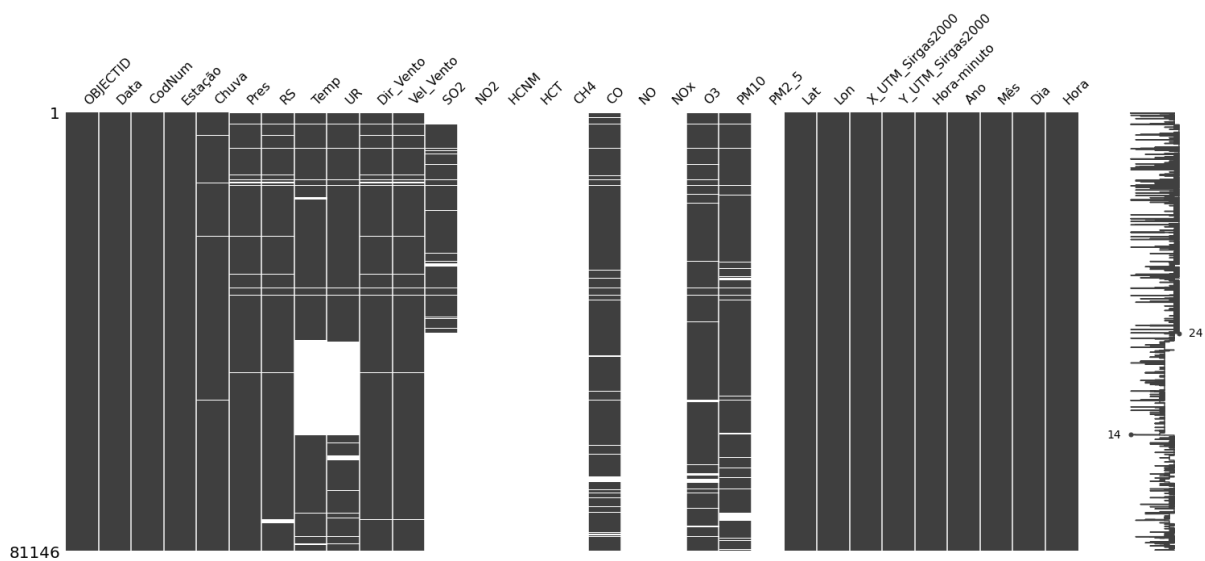


Figura 14 grau de preenchimento da estação do Centro

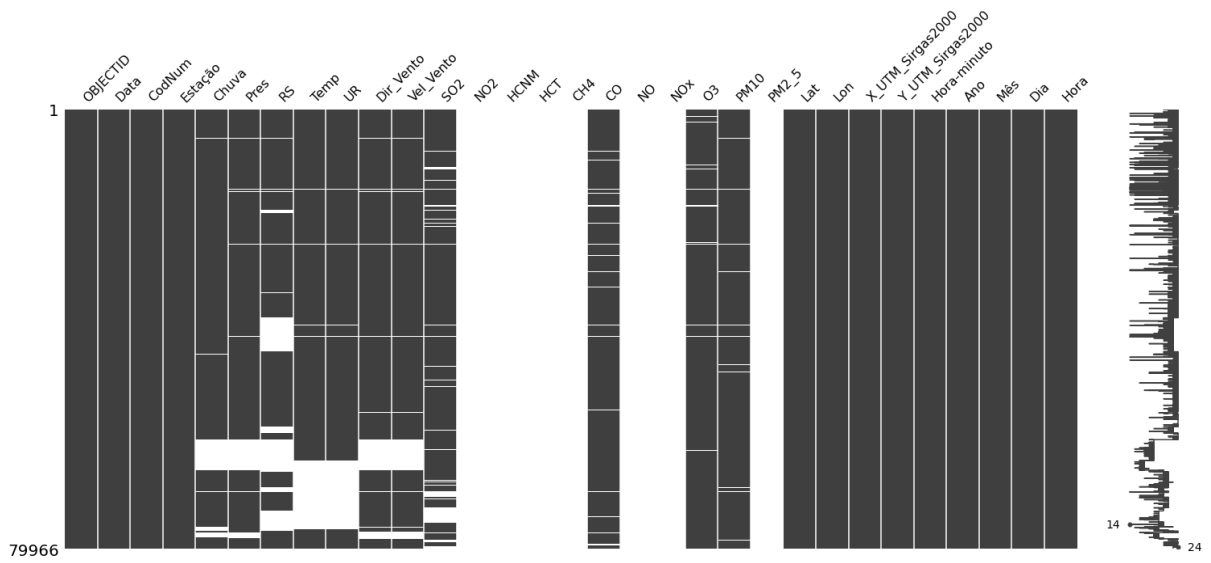


Figura 16 grau de preenchimento da estação do Copacabana

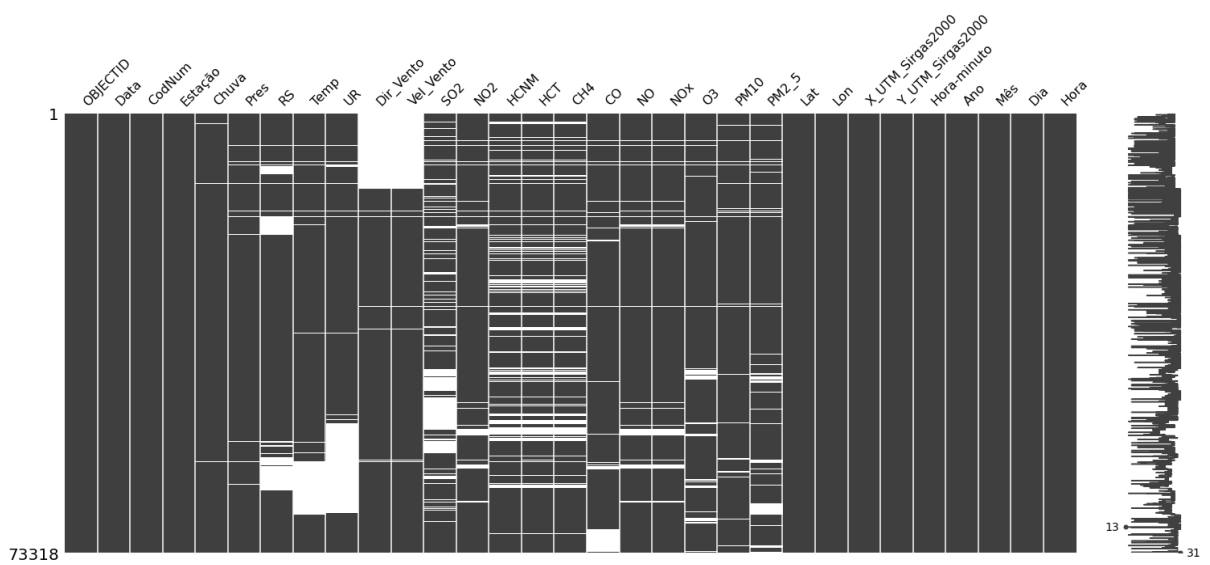


Figura 17 grau de preenchimento da estação do Irajá

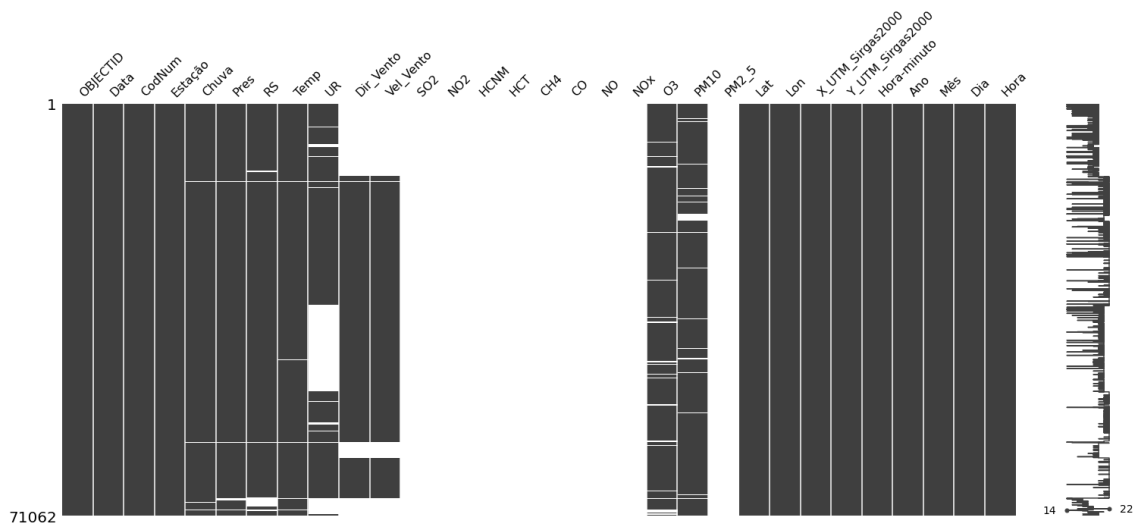


Figura 18 grau de preenchimento da estação de Pedra de Guaratiba

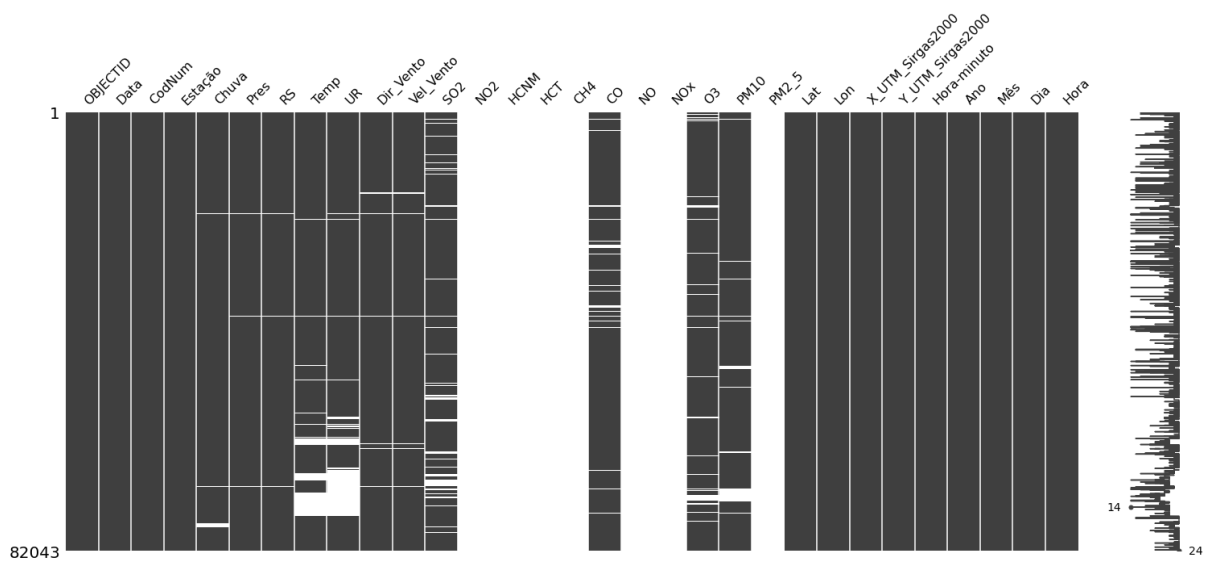


Figura 19 grau de preenchimento da estação de São Cristóvão

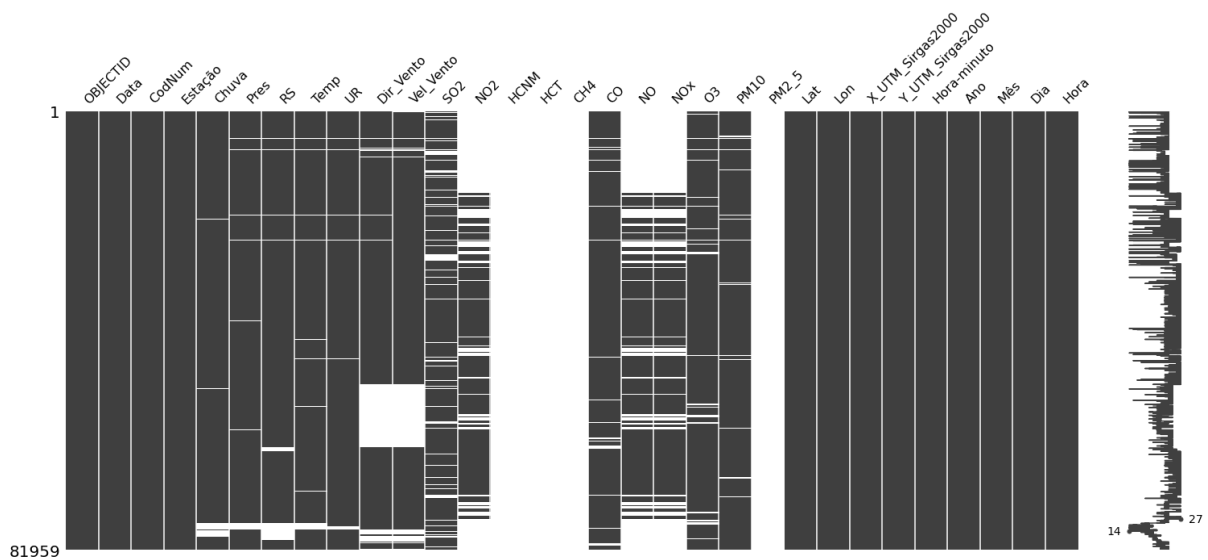


Figura 20 grau de preenchimento da estação da Tijuca

### 5.3 Correlação entre dados de estações adjacentes

Esta etapa da análise tem por objetivo verificar a hipótese de que estações geograficamente mais próximas têm dados mais correlacionados. O intuito é buscar uma justificativa para o tratamento de valores ausentes utilizados nos dados deste trabalho (preenchimento com a média ponderada dos valores para a variável encontrados nas três estações mais próximas). Os mapas de calor representados nas figuras representam as correlações de Spearman<sup>[54]</sup>, para cada variável, entre as diferentes estações. Um valor de 1 corresponde a uma correlação perfeita, um valor de -1 corresponde a uma correlação inversa.

### 5.3.1 Chuva

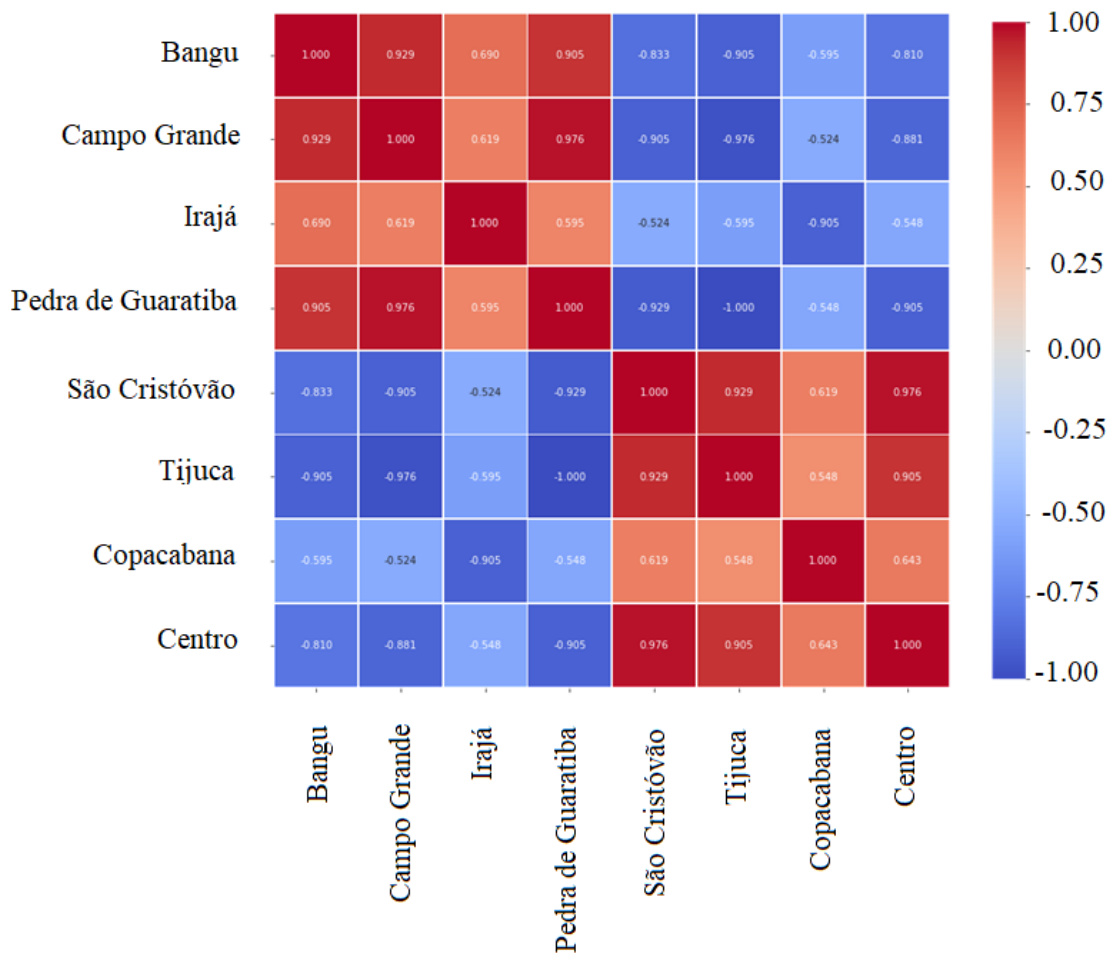


Figura 21 correlação entre a chuva nas oito estações

Como esperado, fica evidente pela cor vermelha no mapa de calor, que as estações mais próximas, como a do cluster formado por São Cristóvão, Tijuca, Copacabana e Centro, têm uma maior correlação de dados relacionados à variável Chuva, desta forma o trabalho pode seguir com o método de substituição de valores ausentes proposto, ao menos para esta variável.

### 5.3.2 Pressão atmosférica

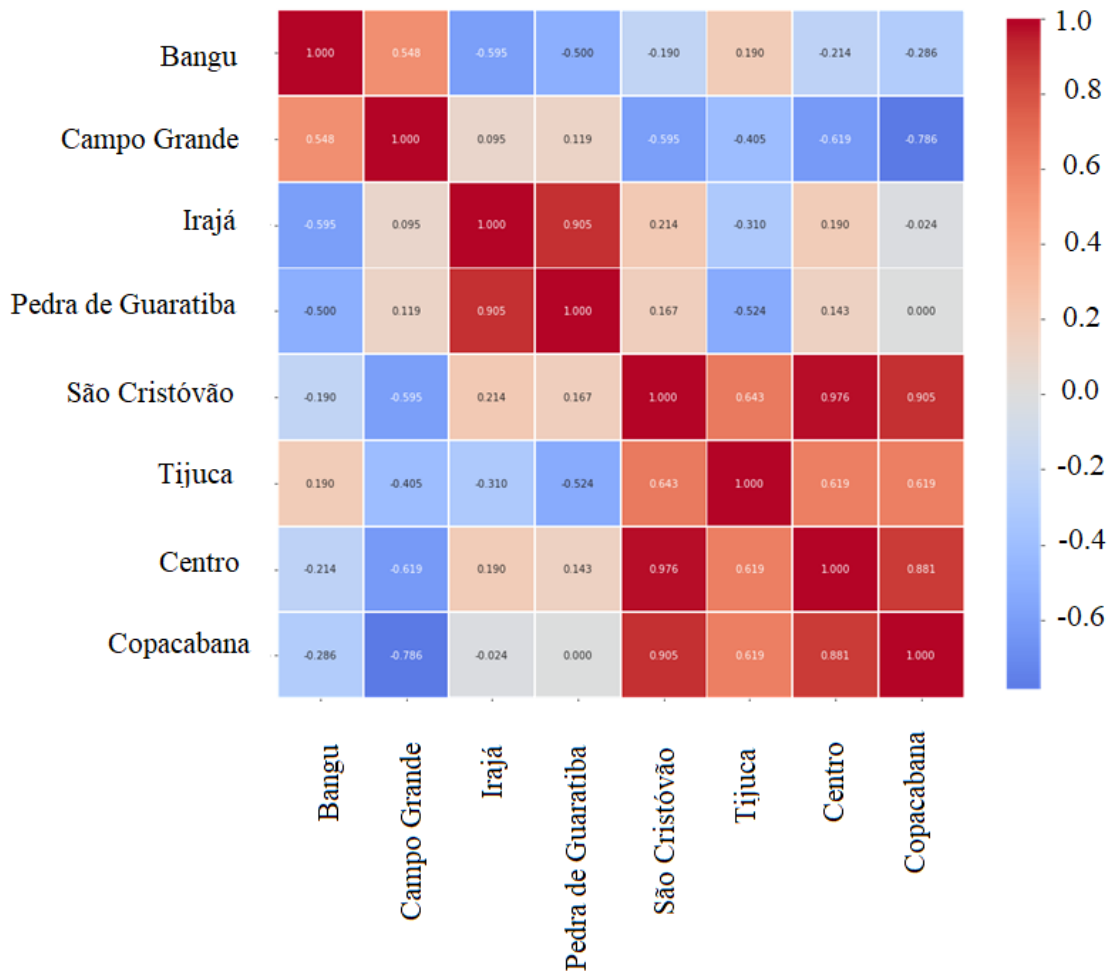


Figura 22 correlação entre a pressão atmosférica nas oito estações

Copacabana e Centro têm uma maior correlação de dados relacionados à pressão, desta forma o método de preenchimento de dados ausentes será adotado também para esta variável.

### 5.3.3 Radiação solar

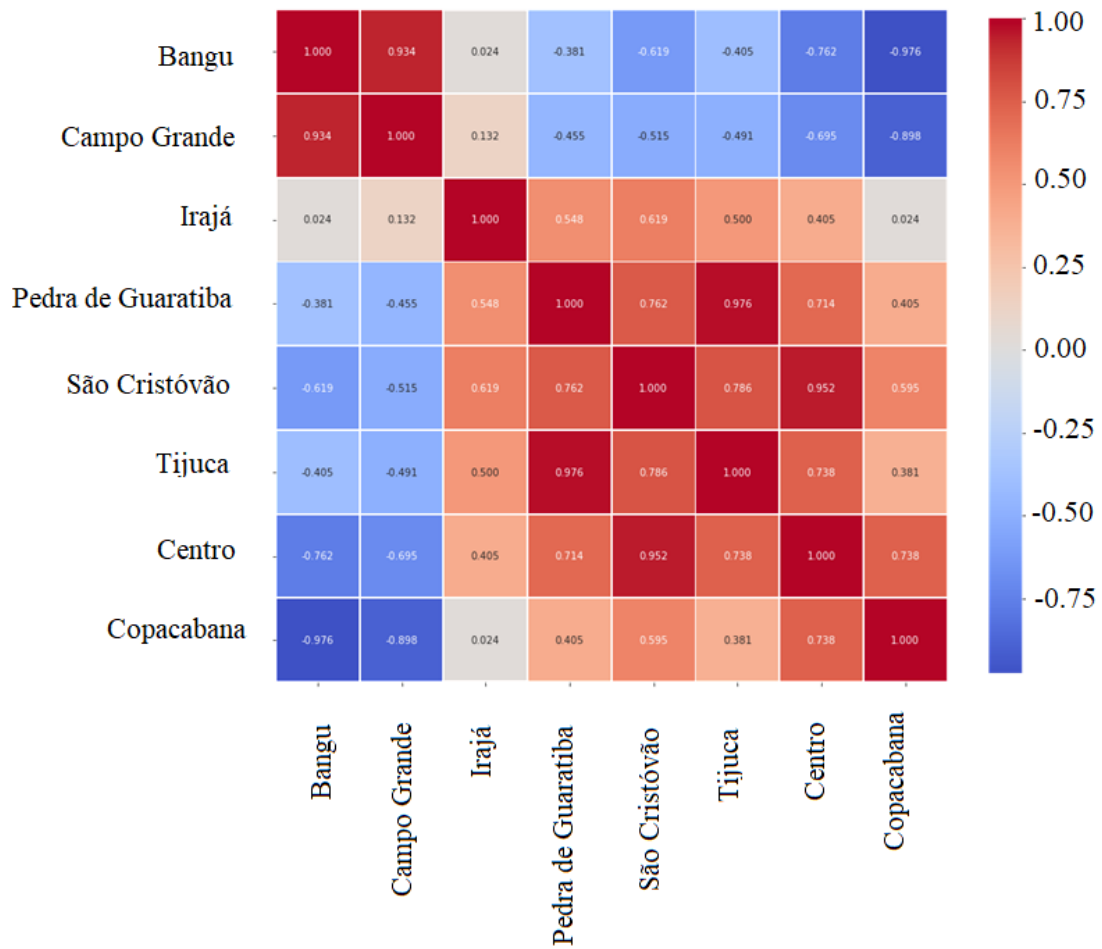


Figura 23 correlação entre a radiação solar nas oito estações

Novamente, as estações mais próximas demonstram ter dados de maior correlação pela análise do mapa de calor, logo para esta variável também será utilizado o método proposto para o preenchimento de dados ausentes.

### 5.3.4 Temperatura

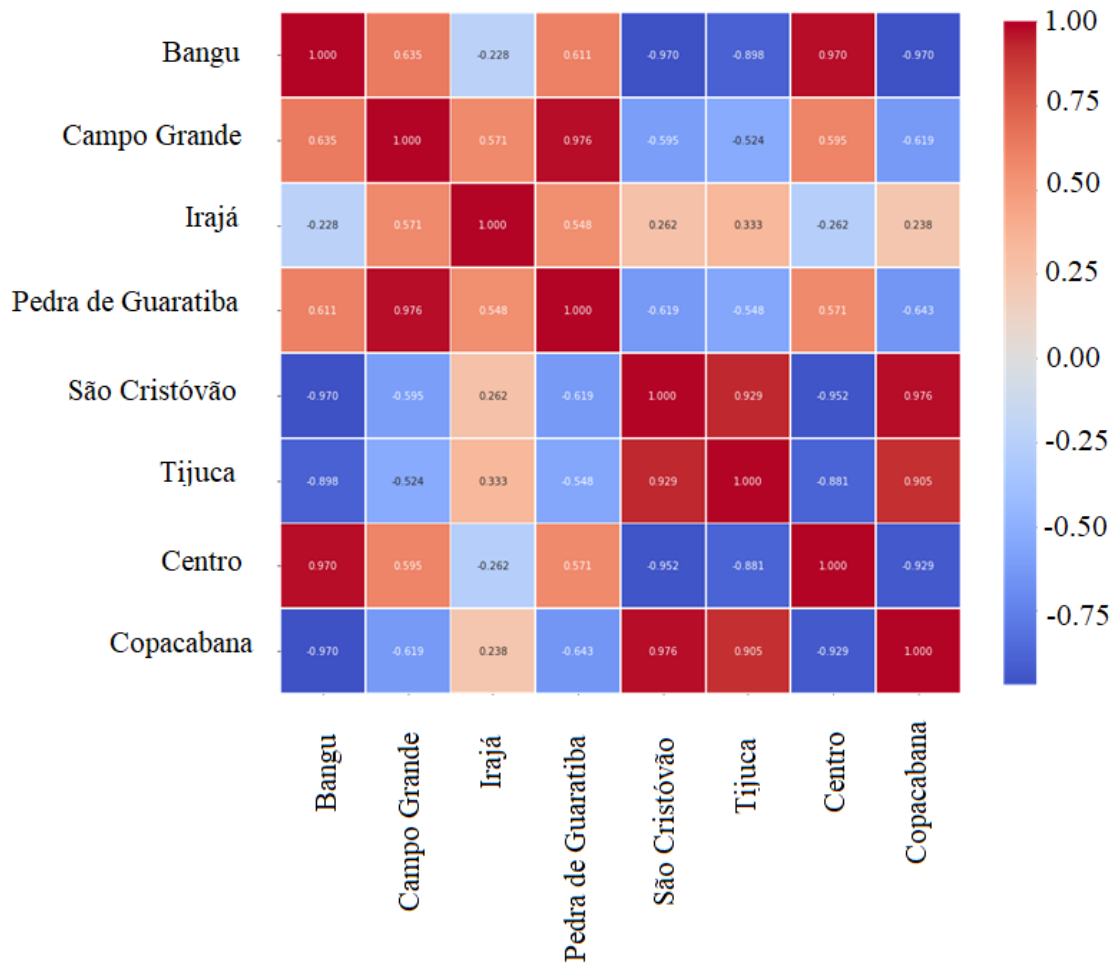


Figura 24 correlação entre a temperatura nas oito estações

Para esta variável, somente as estações imediatamente adjacentes parecem ter uma correlação considerável, como a vista entre os dados da estação de São Cristóvão e da Tijuca.

### 5.3.5 Umidade relativa

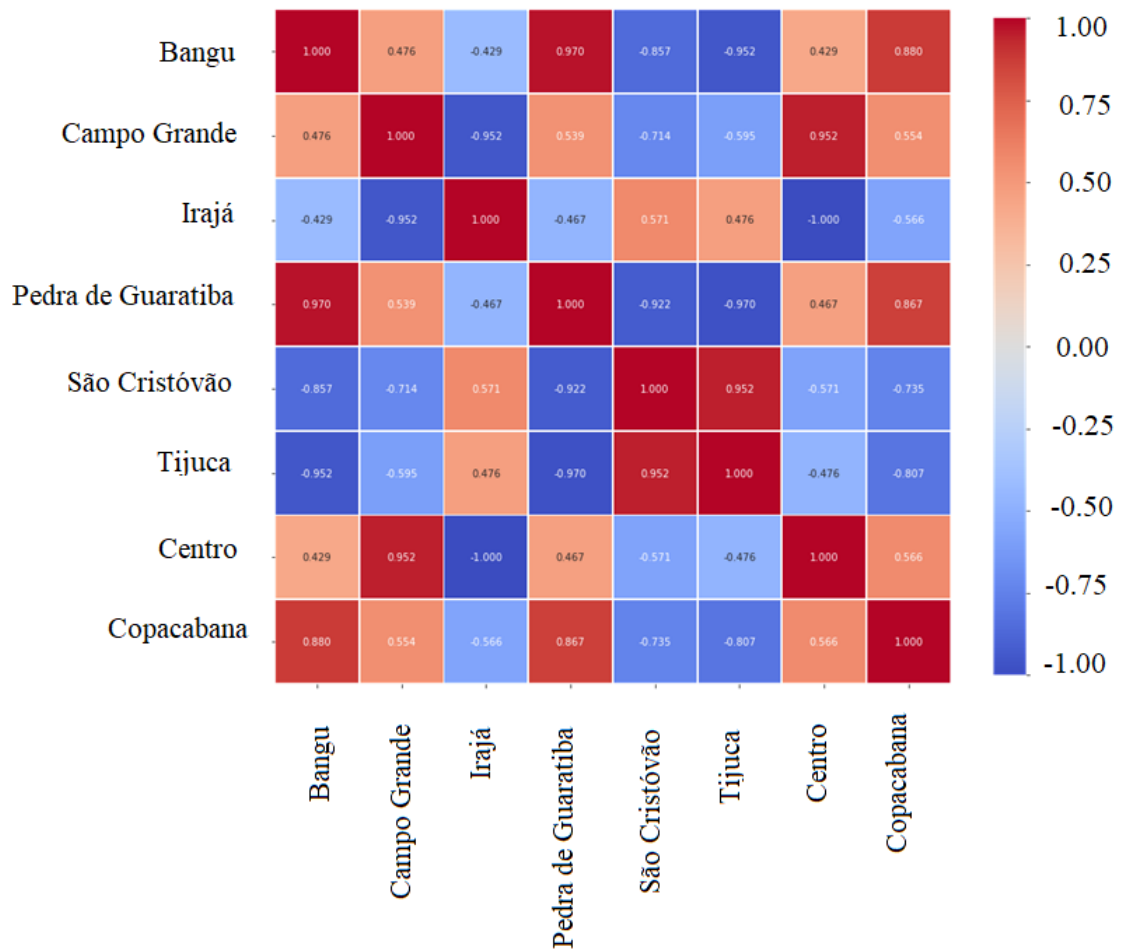


Figura 25 correlação entre a umidade relativa nas oito estações

Assim como para a temperatura, para a umidade relativa somente as estações imediatamente adjacentes parecem ter uma correlação considerável, como a vista novamente entre os dados da estação de São Cristóvão e da Tijuca.

### 5.3.6 Velocidade do vento

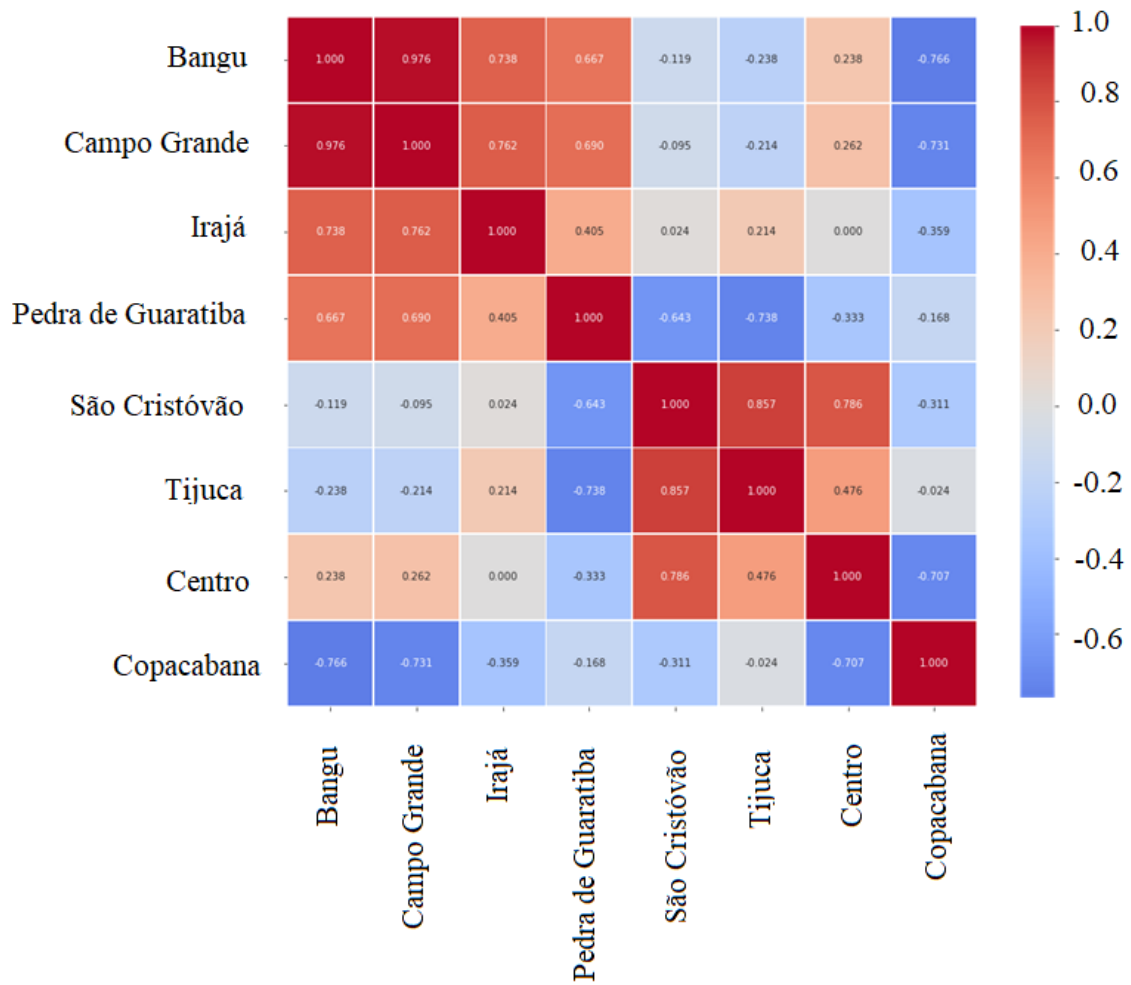


Figura 26 correlação entre a velocidade do vento nas oito estações

Para esta variável, também há uma correlação entre os dados das estações mais próximas, o que fica evidente ao serem observados os valores nos mapas de calor para os dados de São Cristóvão, Tijuca e Centro.

### 5.3.7 Dióxido de enxofre (SO<sub>2</sub>)

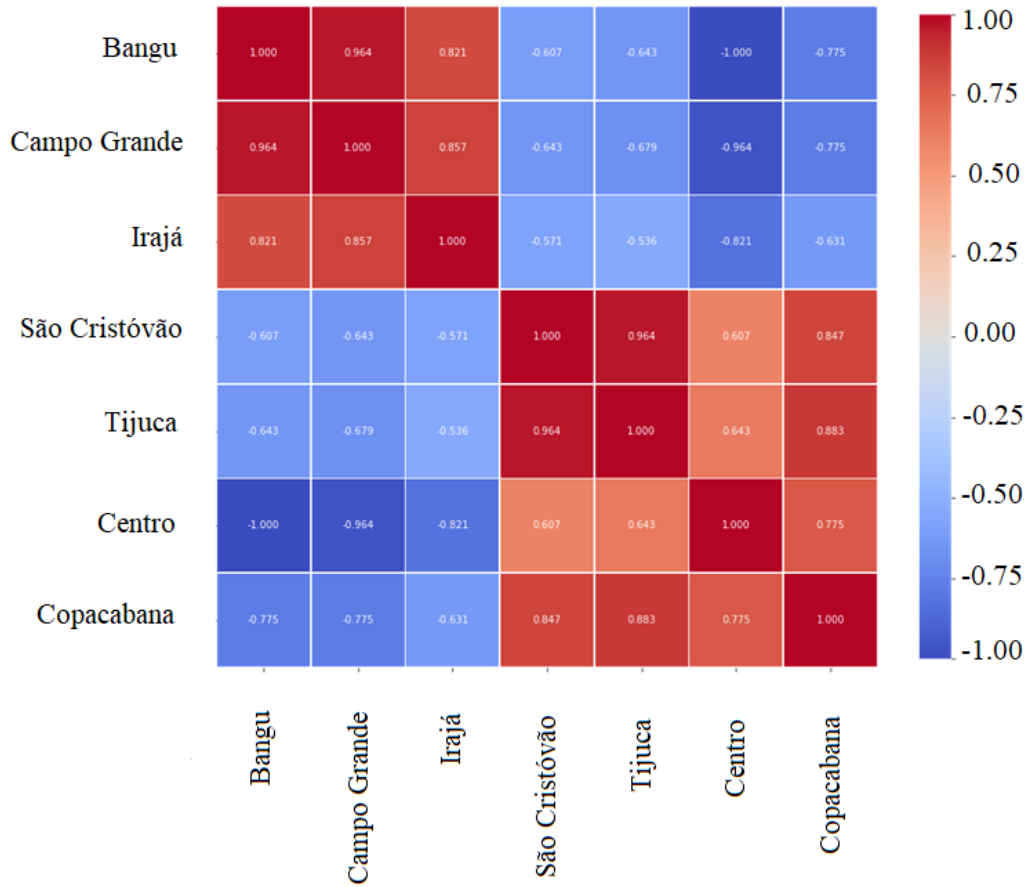


Figura 27 correlação entre o SO<sub>2</sub> nas oito estações

Para esta variável em especial, é evidente pela análise do mapa de calor, que as estações adjacentes têm um valor correlacionado, como as estações de São Cristóvão, Tijuca, Centro e Copacabana.

### 5.3.8 Monóxido de carbono (CO)

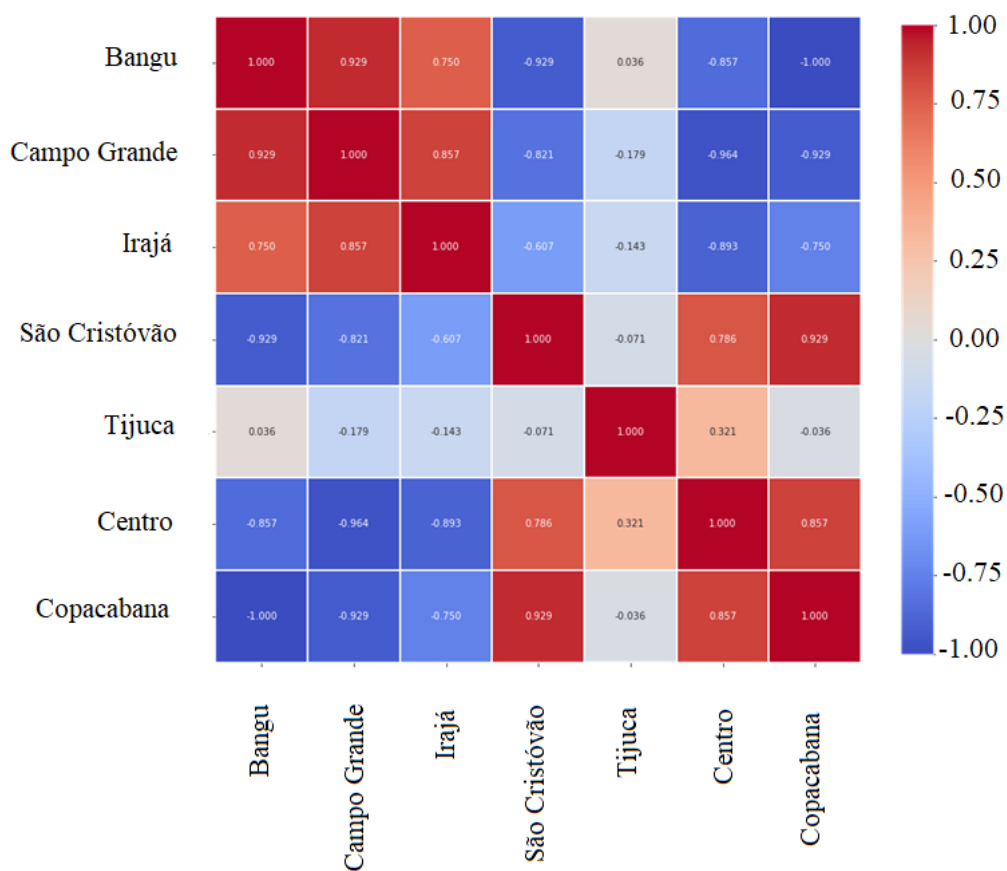


Figura 28 correlação entre o CO nas oito estações

Para esta variável em específico a correlação parece não ser tão evidente quanto nas outras, desta forma, a utilização do método de preenchimento dos dados ausentes proposto, poderá não fornecer os melhores resultados.

### 5.3.9 Ozônio ( $O_3$ )

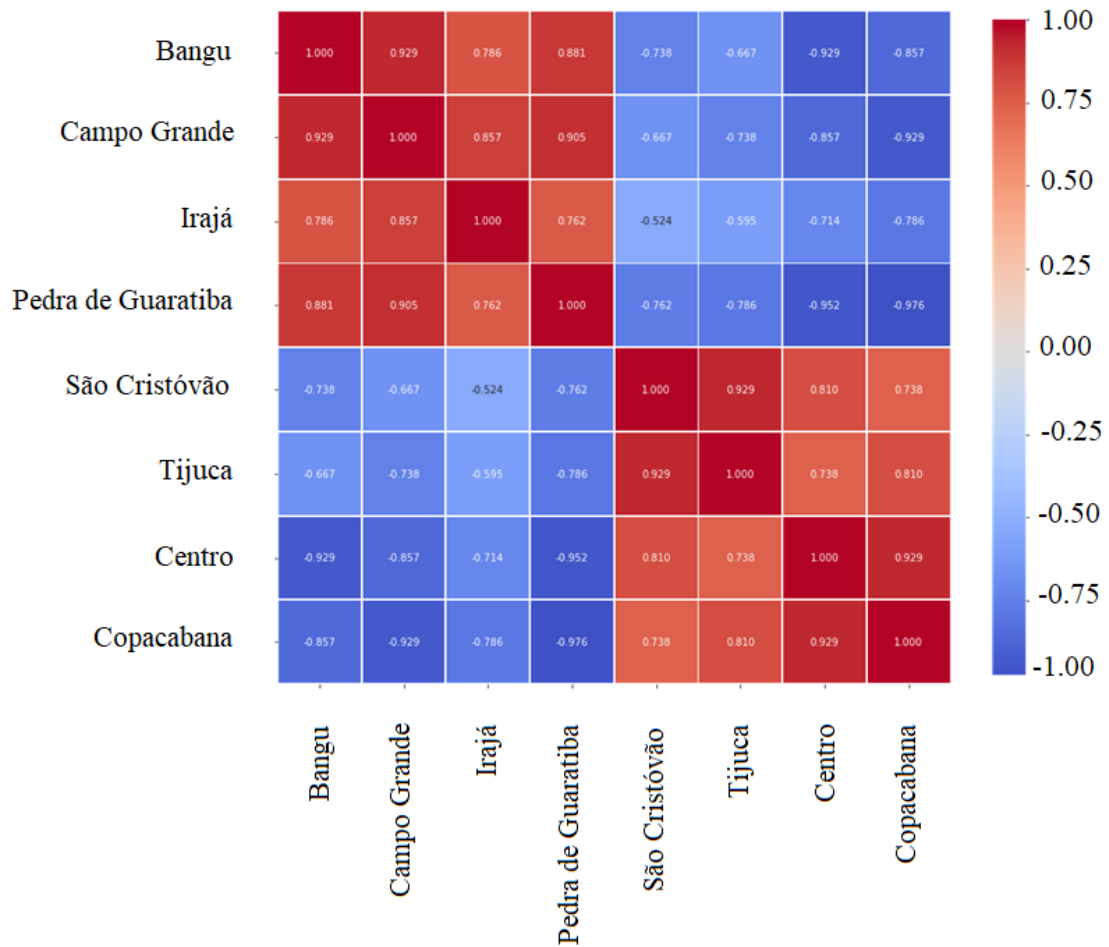


Figura 29 correlação entre o  $O_3$  nas oito estações

Para a variável ozônio, parece novamente haver uma forte correlação entre os dados de estações adjacentes, como mostra o mapa de calor para os dados de São Cristóvão, Tijuca, Centro e Copacabana.

### 5.3.10 Material Particulado ( $PM_{10}$ )

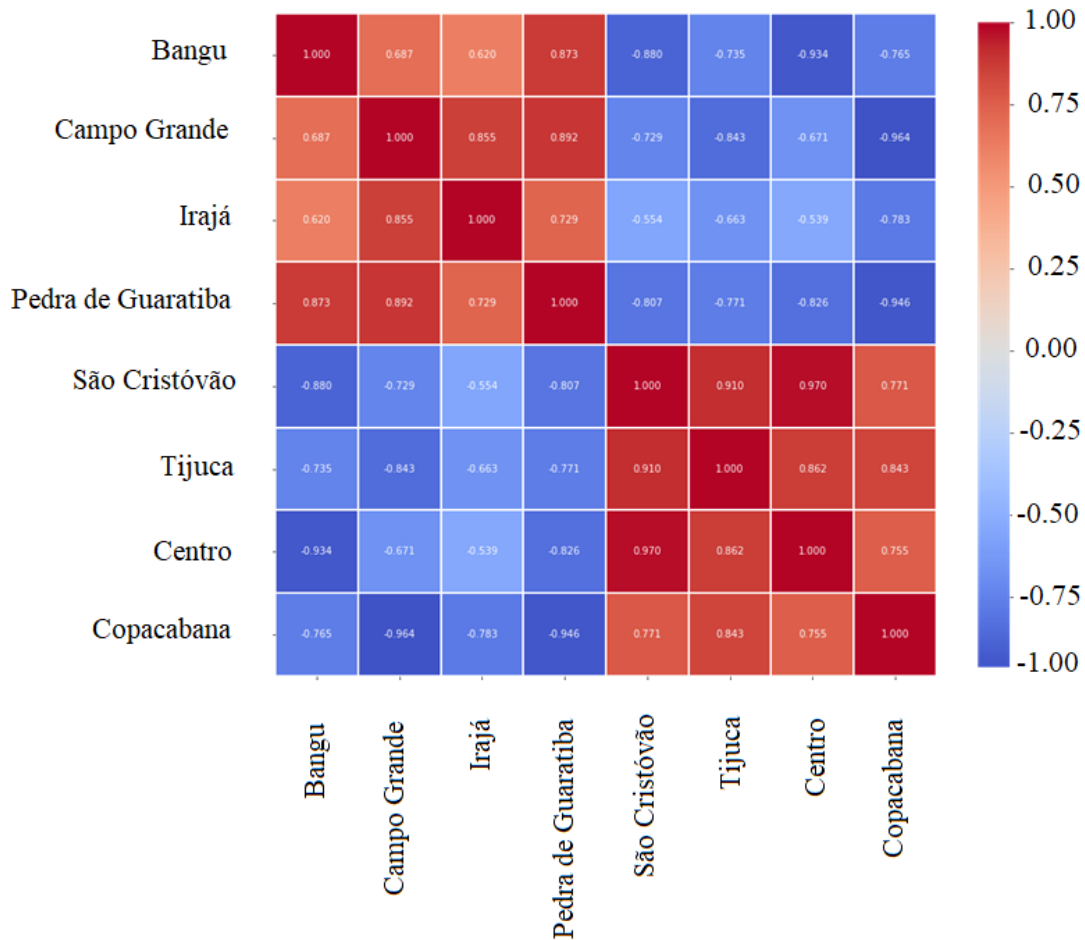


Figura 30 correlação entre o  $PM_{10}$  nas oito estações

Por fim, para a variável  $PM_{10}$ , parece novamente existir uma correlação entre os dados de estações adjacentes, pela análise do mapa de calor para os dados de São Cristóvão, Tijuca, Centro e Copacabana. Desta forma, a metodologia de preenchimento de valores ausentes proposta, é justificada também para esta variável.

## 6 Resultados e discussão

### 6.1 Testes preliminares

Nos testes a seguir, explicados no item 3.8, foram utilizados apenas os dados das estações de São Cristóvão, Tijuca, Copacabana e Centro. Foi considerado que as demais estações se encontram geograficamente muito distantes, para que o método utilizado para preenchimento de dados ausentes seja aplicado à elas. Em todos os testes foi utilizada a mesma taxa de aprendizagem (0.0005) e o mesmo otimizador (ADAM).

A tabela 6 resume os resultados obtidos nesta etapa, comentados com maior detalhe adiante. As métricas utilizadas para avaliar o desempenho de cada rede neural foram:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_i^*)^2} \quad (30)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y_i^*| \quad (31)$$

$$R^2 = 1 - \frac{\sum (y_i - y_i^*)^2}{\sum (y_i - \bar{y}_i)^2} \quad (32)$$

Onde  $y_i^*$  é a concentração do poluente,  $y_i$  é a concentração prevista e  $n$  é o número de registros do conjunto de dados de teste.

Optou-se pela raiz do erro médio quadrático (RMSE), pois esta métrica tende a dar maior peso a erros maiores, uma vez que a diferença entre o valor real e o previsto é elevada ao quadrado, antes que seja feita a média dos erros <sup>[55]</sup>. Esta característica é particularmente interessante para a avaliação dos resultados neste trabalho, em que se deseja obter uma previsão mais próxima possível do valor real. Por uma questão de interpretação dos resultados obtidos, serão avaliados os erros médios absolutos (MAE) fornecidos pelos modelos. O Coeficiente de determinação ( $R^2$ ) é uma métrica que dá ideia de quanto o modelo explica a variância encontrada nos dados. Caso o  $R^2$  seja zero

ou negativo, as previsões geradas pelo modelo não são melhores que a média da variável dependente.

**Tabela 6** resumo dos resultados obtidos

	São Cristóvão	Tijuca	Copacabana	Centro
tamanho do <i>batch</i> de dados	64	128	32	32
arquitetura da rede neural	RNN	GRU	RNN	GRU
número de nós na camada intermediária	32	128	64	64
duração do treinamento (épocas)	150	150	150	150
número de camadas intermediárias	1	1	1	1
normalização de <i>batch</i> e dropout	sem alterações	sem alterações	sem alterações	sem alterações

### 6.1.1 Tamanho do *batch* de dados

Neste teste, para todas as configurações estudadas, foi observado que a métrica  $R^2$  retornou valores negativos, o que na prática pode significar que as previsões do modelo são menos assertivas que a média de cada variável como previsão. Todavia, conforme visto em Spiess *et al.*<sup>[56]</sup>, existe a possibilidade de que esta métrica não se ajuste corretamente a modelos não-lineares, como as redes neurais utilizadas neste trabalho.

Para o modelo treinado com os dados de São Cristóvão, o melhor tamanho para o *batch* de dados pareceu ser 64. Nesta primeira etapa, já foi possível perceber que o modelo forneceu resultados melhores para algumas variáveis. Isto ficou mais evidente quando foi feita a comparação entre os RMSE e MAE com as respectivas médias, para cada variável. Dentre os poluentes, os melhores resultados foram os da variável  $PM_{10}$  cujo valor médio é 34.1 e o RMSE é 17.0 e MAE de 13.83 no melhor modelo. As demais variáveis forneceram RMSE e MAE próximos às médias.

O modelo treinado com os dados da estação da Tijuca teve o melhor desempenho com *batch* de tamanho 128. Utilizando o mesmo critério visto anteriormente, observou-se que o modelo tem seu melhor desempenho no fornecimento de previsões para  $NO$ ,  $NO_2$  e  $NO_x$ , considerando a proporção dos RMSE e MAE

obtidos, comparados com os valores médios de cada variável. Na estação anterior, estes dados não estavam originalmente preenchidos e não houve um bom desempenho do modelo na previsão destas variáveis. Dentre os demais poluentes, há um desempenho fraco na previsão de SO<sub>2</sub>, que tem RMSE de 5.41 e MAE de 4.47 no melhor modelo, e uma média de 3.83.

Para o modelo treinado com os dados da estação de Copacabana, o tamanho ótimo para o *batch* foi 32, o que indica que não houve a princípio uma influência notável do tamanho do *batch* no caso estudado, uma vez que para as estações anteriores os melhores resultados foram obtidos com *batches* de 64 e 128. Até então, para esta estação o modelo forneceu os melhores resultados para PM<sub>10</sub> com RMSE de 12.76 e MAE 10.17 no melhor modelo, tendo um valor médio 33.35.

Assim como para a estação de Copacabana, para a estação do Centro os melhores resultados foram obtidos com *batch* de tamanho 32. Os resultados desta estação foram os melhores em relação à previsão da variável SO<sub>2</sub>, com RMSE de 5.24 e MAE de 4.33 no melhor cenário. Inicialmente as previsões fornecidas para as variáveis NO<sub>x</sub>, NO<sub>2</sub> e NO, que não estavam presentes originalmente neste conjunto, se encontram na mesma faixa de RMSE e MAE das previsões das outras estações, indicando que a estratégia de preenchimento dos valores ausentes não teve um impacto negativo.

### **6.1.2 Arquitetura da rede neural**

Considerando as métricas RMSE e MAE, o modelo treinado com os dados da estação de São Cristóvão, foi ligeiramente melhor com a rede neural do tipo recorrente (RNN), que foi mais assertiva para 6 das 12 variáveis previstas. As mesmas métricas indicam também que a rede neural do tipo GRU obtém um desempenho próximo ao da melhor rede, sendo a que melhor previu 4 variáveis das 12. O modelo LSTM foi o melhor para 3 das 12 variáveis. Observando o tempo de treinamento das redes neurais, é possível notar que a rede neural recorrente, que obteve os menores RMSE e MAE em 6 ocasiões, não é a rede cujo treinamento foi o mais rápido, sendo superada pela rede LSTM. Além disso, assim como visto em Namini *et al.* <sup>[57]</sup>, é notável que a rede BI-LSTM tem um tempo de treinamento mais lento que as demais redes. Por fim, usando

como parâmetro a rede neural de melhor desempenho (RNN), com os dados da estação de São Cristóvão, já foi possível observar melhorias nos desempenhos das previsões fornecidas pelo modelo. Exceto pelas previsões fornecidas para a variável CO, todas elas obtiveram RMSE e MAE menores.

O melhor modelo treinado com os dados da estação da Tijuca foi o que utilizou a rede GRU, e foi mais assertivo para 5 das 12 variáveis, de acordo com os RMSE e MAE dos testes. A rede neural recorrente simples e a rede LSTM tiveram melhor desempenho nas previsões de 3 das 12 variáveis. Com os dados desta estação, a rede BI-LSTM teve novamente um desempenho inferior, tendo fornecido as melhores previsões para apenas duas variáveis. Observando o tempo de treinamento, as redes RNN, BI-LSTM e GRU obtiveram tempos significativamente próximos. Os tempos de treinamento de rede LSTM foram ligeiramente menores, indicando que caso exista uma massa de dados maior, este modelo pode ser mais adequado, uma vez que não haja grandes disparidades entre os RMSE e MAE de cada modelo.

Utilizando os dados da estação de Copacabana, o melhor modelo treinado, de acordo com as métricas RMSE e MAE, utilizou a rede RNN, que foi a melhor para a previsão de 7 das 12 variáveis. Para esta rede o número ótimo de neurônios por camada parece ser novamente 32 ou 64. A segunda melhor rede foi a LSTM, que forneceu previsões mais assertivas para 3 das 12 variáveis. A rede GRU foi a melhor em dois casos, ao passo que a rede BI-LSTM forneceu a melhor previsão para apenas uma variável. As redes RNN, GRU e LSTM tiveram tempos de treinamento muito próximos, ao passo que novamente a rede BI-LSTM obteve um tempo médio de treinamento de cerca de 40% maior que o tempo médio de treinamento das outras redes, sem ganho significativo no desempenho do modelo.

Para a estação do Centro, o modelo com as previsões mais assertivas, considerando as métricas RMSE e MAE, foi a rede neural do tipo GRU, que foi a melhor para 6 das 12 variáveis. O segundo melhor desempenho foi o da rede neural do tipo RNN, que foi mais assertiva na previsão de 4 das 12 variáveis do conjunto de dados. A rede LSTM foi a melhor em 2 ocasiões, ao passo que a rede do tipo BI-LSTM foi novamente a de desempenho mais fraco, sendo a melhor na previsão de apenas uma variável. A rede neural do tipo LSTM foi novamente a mais rápida, o que contradiz o

visto em Chung *et al.* [58] e Yang *et al.* [59], que dizem que a rede GRU tende a ser mais rápida que a LSTM, devido a possuir menos hiperparâmetros para serem treinados, sendo por isto computacionalmente mais eficiente. Uma possível explicação é que o tamanho do conjunto de dados, da magnitude de milhares de registros, não é grande o suficiente para resultados semelhantes aos mostrados nestes trabalhos, onde fica clara a maior rapidez da rede GRU em relação a LSTM. Os mesmos trabalhos mostram que a rede LSTM fornece resultados mais assertivos com grandes volumes de dados, ao passo que a rede do tipo GRU é melhor com menores volumes de dados, o que corrobora com os resultados obtidos no presente trabalho.

Ao fim desta etapa de testes, foi observado que, com a configuração atual do problema e com os dados disponíveis, as redes RNN e GRU obtiveram desempenhos equiparáveis. No entanto, a rede neural GRU aparentou ter treinamento ligeiramente mais rápido.

### ***6.1.3 Número de nós na camada intermediária***

O modelo treinado com os dados da estação de São Cristóvão teve melhor desempenho com 32 nós na camada intermediária. Levando em conta esta configuração, é possível notar uma melhora nas previsões para o poluente O<sub>3</sub>, com RMSE de 18.24 e MAE de 15.32. No primeiro experimento o melhor modelo forneceu resultados com RMSE de 19.62 e MAE de 15.32. No segundo experimento, para esta estação, o RMSE das previsões foi de 19.63 e o MAE 15.85. Os demais poluentes também obtiveram melhoras, em relação aos testes anteriores, no entanto, como dito anteriormente, a melhor configuração não necessariamente é a mesma que a configuração ótima para o poluente O<sub>3</sub>.

Para a estação da Tijuca, o melhor modelo foi obtido com 128 neurônios por camada, que conseguiu melhor desempenho para 4 das 12 variáveis que compõem o conjunto de dados. Para esta configuração, a princípio não parece ter havido uma melhora notável no desempenho em relação à etapa anterior. Verificando o caso da variável SO<sub>2</sub>, foram mantidos os mesmos RMSE de 5.45 e MAE de 4.46, conseguidos no melhor desempenho na etapa anterior, que também teve cerca de 2 minutos para encerrar o treinamento do modelo.

Para o modelo treinado com os dados da estação de Copacabana, o melhor desempenho foi alcançado com 64 nós na camada intermediária, que foi o melhor para 4 das 12 variáveis do conjunto de dados. Avaliando a duração dos treinamentos, não houve notável discrepância com relação ao ocorrido nas demais estações, com os tempos estando em torno dos 90 segundos para cada cenário testado.

O modelo treinado com os dados da estação do Centro teve melhor desempenho com 32 ou 64 nós na camada intermediária. Dentre os modelos de melhor desempenho, o de 32 neurônios na camada intermediária foi ligeiramente mais rápido com 109.1 segundos de treinamento, enquanto o de 64 neurônios na camada intermediária teve duração total de 124.1 segundos de treino.

Desta forma, considerando os resultados dos testes para as quatro estações, o melhor modelo teve como parâmetro 64 neurônios na camada intermediária.

#### ***6.1.4 Duração do treinamento***

Para todas as estações, o melhor desempenho foi alcançado com 150 épocas de treinamento. O comportamento do treinamento com maiores durações mostra que em torno da época 150 é alcançado um patamar, no qual os resultados não melhoram não importando o quanto dure o treinamento, o que é um comportamento típico do problema de dissipação do gradiente. Desta forma, é possível que com medidas para mitigar este problema, o treino possa ter sua duração estendida, aprimorando o desempenho do modelo.

#### ***6.1.5 Número de camadas intermediárias***

Tradicionalmente as redes neurais possuem três tipos de camadas: de *input*, de *output* e intermediárias. As camadas intermediárias podem ser de diferentes tipos, de acordo com a finalidade. Neste trabalho, foram utilizadas camadas do tipo recorrente.

De acordo com Pascanu *et al.* <sup>[60]</sup>, modelos com redes mais profundas podem ser exponencialmente mais eficientes em representar determinadas funções que modelos com menos camadas. Segundo Hermans *et al.* <sup>[61]</sup>, ao adicionarem-se mais camadas a uma rede neural, é introduzida uma hierarquia às diversas camadas do modelo, de forma

que cada camada processará uma parte do problema a ser resolvido, o que melhora o resultado final.

Para o modelo treinado com os dados de São Cristóvão, de acordo com as métricas RMSE e MAE houve uma predominância de melhores resultados entre os modelos de menos camadas, com o modelo de 1 camada tendo obtido melhores resultados em 6 das 12 variáveis. Para apenas duas variáveis um menor número de camadas não prevaleceu,  $PM_{10}$ , onde o melhor desempenho ocorreu com 3 camadas, e CO, que teve uma das melhores performances com 4 camadas. Os resultados exibidos não vão de encontro ao que é esperado de acordo com os trabalhos de Pascanu *et al.* e Hermans *et al.*, onde é relatado que há uma tendência de que modelos com mais camadas forneçam melhores resultados. Os resultados vistos nestes trabalhos podem, no entanto terem sido diferentes devido às especificidades encontradas nos dados. Outras possíveis explicações para o ocorrido são: aumentar o número de camadas poderia piorar o problema da dissipação do gradiente; o número alto de camadas e neurônios poderá levar ao sobreajuste do modelo; o número de épocas de treinamento pode ser alto, levando em consideração o tamanho do conjunto de dados.

Os modelos treinados com os dados da estação da Tijuca, obtiveram melhores RMSE e MAE, com 1 camada, sendo a melhor composição para 9 das 12 variáveis. Como já dito anteriormente, os resultados podem ter sido influenciados pelo número de épocas de treinamento ter sido o mesmo para todas as configurações de número de camadas. Supõe-se que um modelo de mais camadas demandaria uma maior duração do treinamento para chegar ao melhor desempenho, todavia aumentar a duração do treino poderia levar a o sobreajuste do modelo ou agravar o problema da dissipação do gradiente.

Com os dados da estação de Copacabana, a exemplo do ocorrido nas duas estações anteriores, novamente o melhor modelo foi o de uma camada intermediária, que obteve as melhores métricas em 7 das 12 variáveis. No entanto, assim como ocorrido anteriormente não houve grande disparidade entre os resultados dos modelos. Mais uma vez, houve uma tendência de o modelo de uma camada ter seu treinamento encerrado mais rapidamente que os demais, o que pode ser explicado por um menor número de hiperparâmetros a serem ajustados.

Assim como aconteceu com os dados das outras estações, para a estação do Centro, o modelo de uma camada intermediária se saiu melhor nas previsões, fornecendo valores mais assertivos para 10 das 12 variáveis.

Para todas as estações, é necessário observar que em trabalhos futuros os modelos de mais camadas poderão ter um melhor desempenho caso sejam treinados por com mais épocas de treino que o modelo mais simples.

### **6.1.6 Normalização de batch e dropout**

De acordo com Srivastava *et al.* <sup>[62]</sup>, em determinadas situações a técnica de *dropout* pode melhorar o desempenho de uma rede neural ao prevenir o sobreajuste da rede durante seu treinamento. Outra possível técnica de eficácia comprovada é a regularização de *batch* <sup>[63]</sup>, que segundo o autor, dentre outros benefícios, também pode auxiliar na prevenção da dissipação do gradiente, levando geralmente a melhores resultados. Desta forma, nesta etapa foram testadas combinações de redes neurais com *dropout* e normalização de *batch*.

Para o modelo treinado com os dados de São Cristóvão, de acordo com as métricas RMSE e MAE, nenhum dos modelos, em que houve a utilização da técnica de normalização de *batch*, alcançou o melhor desempenho, para pelo menos uma das variáveis. De acordo com Coojimans *et al.* <sup>[64]</sup>, em redes neurais do tipo recorrente a utilização desta técnica pode prejudicar o desempenho, uma vez que a normalização não levaria em consideração a recorrência contida neste tipo de rede. No entanto, conforme Wang *et al.* <sup>[65]</sup>, em determinadas situações, as redes do tipo GRU, podem se beneficiar da utilização de normalização de *batch*. É possível também, que a utilização da normalização de *batch* diminua a quantidade de épocas para treinar o modelo, e que o excesso de treinamento tenha prejudicado o desempenho nestes casos. Para esta estação, o melhor desempenho foi o do modelo sem regularização com *dropout*, que foi o melhor para 7 das 12 variáveis. De acordo com Ohlsson <sup>[66]</sup>, redes neurais pouco profundas geralmente se beneficiam menos da técnica de *dropout*, o artigo cita ainda que uma vez que esta regularização é utilizada, pode ser necessário aumentar o tempo de treinamento para otimizar os resultados, o que pode explicar a menor eficácia da utilização de *dropout* nesta rede. Outro ponto a ser destacado é que o treinamento

ocorreu mais rápido com o uso de *dropout*, sem perda considerável na qualidade do resultado.

Dentre os modelos treinados com os dados da estação da Tijuca, em nenhuma das configurações a normalização de *batch* melhorou o desempenho dos modelos. Como dito anteriormente, é possível que a natureza recorrente da rede neural utilizada no trabalho não seja compatível com a técnica. É possível também que o modelo que utiliza normalização em *batch* necessite de menos épocas de treinamento para atingir o valor ótimo. A técnica de *dropout* foi capaz de atingir o melhor resultado para 3 das 12 variáveis, o que indica que também nesta estação, a técnica de *dropout* não resultou em melhora no desempenho do modelo. No entanto, um *dropout* de 20% reduziu a menos da metade o tempo necessário para encerrar o treinamento do modelo, sem perdas consideráveis no desempenho, o que poderia ser útil caso a massa de dados fosse maior.

Assim como para as duas estações anteriores, São Cristóvão e Tijuca, a normalização de *batch* não resultou em um melhor desempenho para o modelo para o modelo treinado com os dados da estação de Copacabana. Neste teste, a rede sem regularização de *dropout* atingiu novamente o melhor desempenho para 7 das 12 variáveis que compõem o conjunto de dados. Todavia em 4 ocasiões o *dropout* de 10% igualou ou melhorou os resultados obtidos, e em outras 4 ocasiões um *dropout* de 15% também melhorou as previsões retornadas pelo modelo. Uma vez que o tempo de treino é diminuído consideravelmente, utilização de *dropout* pode ser uma opção viável caso haja uma massa de dados significativamente maior que a utilizada neste trabalho.

A exemplo do ocorrido nas estações de São Cristóvão, Tijuca e Copacabana, o modelo treinado com os dados da estação do Centro obteve melhor desempenho, para 7 das 12 variáveis, em sua versão sem regularização de *dropout*. Novamente houve uma redução considerável do tempo de treinamento, chegando a um terço do tempo original com a utilização de *dropout* de 20%, sem perda considerável de desempenho nas previsões. A normalização de *batch* não surtiu efeitos positivos visíveis. Uma vez que o modelo sem *dropout* foi o melhor em todos os testes, o modelo ensemble não conterà regularização de *dropout* ou normalização de *batch*. Deve ser lembrado que é possível a variação do número de épocas de treinamento, aumentando-as no caso do *dropout* ou

diminuindo-a no caso da normalização de *batch*, poderá produzir melhoras nos RMSE e MAE obtidos pelos modelos.

## 6.2 Ensemble de modelos

Nesta etapa, é testada a hipótese de que um ensemble de modelos, utilizando as três redes do tipo recorrente (GRU, LSTM e RNN) poderá obter melhor desempenho de que as três redes separadamente. Por razão de simplificação, a melhor configuração de hiperparâmetros e arquitetura obtida para a rede de melhor desempenho (GRU) é replicada para as demais redes (RNN e LSTM).

### 6.2.1 São Cristóvão

**Tabela 7** comparação do desempenho do modelo ensemble com o desempenho das redes simples, utilizando dados da estação de São Cristóvão

coluna	métrica	Ensemble	LSTM	GRU	RNN
Temperatura	RMSE	3,35	3,46	3,32	3,46
NOx	RMSE	24,57	24,83	24,90	24,30
CO	RMSE	0,26	0,26	0,26	0,26
Chuva	RMSE	1,85	1,85	1,90	1,91
Pressão	RMSE	2,56	2,73	2,71	2,96
Velocidade do vento	RMSE	0,60	0,59	0,60	0,60
SO <sub>2</sub>	RMSE	5,13	5,18	5,19	5,25
PM10	RMSE	15,24	15,43	15,48	15,53
O <sub>3</sub>	RMSE	17,86	18,10	18,18	17,91
UR	RMSE	7,28	7,04	6,79	7,19
NO	RMSE	16,84	16,83	16,73	16,37
NO <sub>2</sub>	RMSE	14,94	14,94	15,48	14,58
Temperatura	MAE	2,64	2,71	2,63	2,69
NOx	MAE	19,32	19,75	19,64	18,96
CO	MAE	0,21	0,21	0,20	0,21
Chuva	MAE	1,10	1,16	1,24	1,28
Pressão	MAE	1,99	2,13	2,11	2,32
Velocidade do vento	MAE	0,48	0,48	0,48	0,49
SO <sub>2</sub>	MAE	4,17	4,18	4,25	4,30
PM10	MAE	12,38	12,59	12,52	12,60
O <sub>3</sub>	MAE	14,27	14,50	14,57	14,25
UR	MAE	5,88	5,59	5,36	5,73
NO	MAE	13,18	13,20	13,14	12,69
NO <sub>2</sub>	MAE	11,58	11,65	12,04	11,21
Temperatura	R <sup>2</sup>	0,03	-0,01	0,09	0,06
NOx	R <sup>2</sup>	0,10	0,06	0,15	0,10
CO	R <sup>2</sup>	-1,90	-2,15	-1,71	-2,20
Chuva	R <sup>2</sup>	-6,86	-6,37	-5,82	-6,65
Pressão	R <sup>2</sup>	0,53	0,36	0,46	0,28
Velocidade do vento	R <sup>2</sup>	-3,05	-3,24	-2,87	-3,26
SO <sub>2</sub>	R <sup>2</sup>	-2,30	-2,59	-2,26	-2,20
PM10	R <sup>2</sup>	-0,93	-1,01	-1,07	-0,95
O <sub>3</sub>	R <sup>2</sup>	-0,57	-0,54	-0,51	-0,50

UR	R <sup>2</sup>	-1,08	-0,92	-0,75	-0,80
NO	R <sup>2</sup>	0,06	0,11	0,10	0,16
NO <sub>2</sub>	R <sup>2</sup>	0,27	0,35	0,23	0,36

Conforme os resultados mostrados na tabela acima, foi possível notar que, mesmo com o modelo ensemble, a métrica R<sup>2</sup> apresenta valores muito abaixo do esperado. Como já dito inicialmente, isto pode ser explicado pela baixa adequação da métrica a modelos não-lineares. De acordo com as métricas MAE e RMSE o modelo ensemble é superior aos modelos mais simples para 7 das 12 variáveis.

Na primeira etapa o melhor modelo teve RMSE de 0.26 e MAE de 0.22 para o poluente CO, o modelo ensemble obteve 0.26 e 0.21, perante um valor médio de 0.24 para a variável. Para o poluente SO<sub>2</sub> o melhor modelo de primeira etapa obteve RMSE de 9.19 e MAE de 7.78, no modelo ensemble estes valores caíram para 5.13 e 4.17 respectivamente, perante um valor médio de 8.48. O poluente PM<sub>10</sub> teve RMSE de 17.04 e MAE de 13.83 na primeira etapa, o modelo ensemble obteve RMSE de 15.24 e MAE de 12.38, com um valor médio de 34.13. O poluente teve como resultado RMSE de 18.48 e MAE de 14.80 na primeira etapa, o modelo ensemble teve RMSE de 17.86 e MAE de 14.27, com um valor médio de 23.9. Na primeira etapa o modelo retornou RMSE de 14.97 e MAE de 11.78, ao passo que o modelo ensemble retornou RMSE de 14.94 e MAE de 11.78.

## 6.2.2 Tijuca

**Tabela 8** comparação do desempenho do modelo ensemble com o desempenho das redes simples, utilizando dados da estação da Tijuca

coluna	métrica	Ensemble	LSTM	GRU	RNN
Temperatura	RMSE	4,13	4,38	4,24	4,13
NOx	RMSE	25,97	25,33	26,53	25,64
CO	RMSE	0,20	0,20	0,20	0,21
Chuva	RMSE	1,84	1,84	1,88	1,81
Pressão	RMSE	2,76	2,89	2,66	2,78
Velocidade do vento	RMSE	0,70	0,70	0,72	0,72
SO <sub>2</sub>	RMSE	5,49	5,62	5,51	5,43
PM10	RMSE	16,04	15,85	15,80	16,59
O <sub>3</sub>	RMSE	17,43	18,14	17,63	17,39
UR	RMSE	7,96	8,05	8,14	8,49
NO	RMSE	17,32	16,91	16,70	17,69
NO <sub>2</sub>	RMSE	15,00	15,29	15,30	16,31
Temperatura	MAE	3,28	3,51	3,39	3,29
NOx	MAE	20,71	20,00	21,02	20,45
CO	MAE	0,16	0,16	0,16	0,16

Chuva	MAE	1,06	1,11	1,17	1,05
Pressão	MAE	2,15	2,26	2,07	2,19
Velocidade do vento	MAE	0,55	0,54	0,57	0,56
SO <sub>2</sub>	MAE	4,51	4,60	4,50	4,46
PM10	MAE	12,94	12,62	12,65	13,40
O <sub>3</sub>	MAE	13,81	14,46	13,98	13,79
UR	MAE	6,36	6,46	6,52	6,80
NO	MAE	13,77	13,43	13,20	13,95
NO <sub>2</sub>	MAE	11,61	11,81	11,88	12,91
Temperatura	R <sup>2</sup>	0,00	-0,17	-0,03	0,08
NOx	R <sup>2</sup>	-0,06	0,04	0,08	0,06
CO	R <sup>2</sup>	-2,62	-2,84	-2,09	-2,53
Chuva	R <sup>2</sup>	-8,15	-8,68	-8,27	-8,08
Pressão	R <sup>2</sup>	0,48	0,40	0,58	0,51
Velocidade do vento	R <sup>2</sup>	-4,68	-4,93	-4,60	-3,81
SO <sub>2</sub>	R <sup>2</sup>	-1,72	-1,78	-1,65	-1,69
PM10	R <sup>2</sup>	-1,46	-1,36	-1,25	-1,52
O <sub>3</sub>	R <sup>2</sup>	-0,18	-0,34	-0,18	-0,14
UR	R <sup>2</sup>	-0,90	-0,85	-1,01	-0,86
NO	R <sup>2</sup>	-0,06	0,06	0,07	-0,02
NO <sub>2</sub>	R <sup>2</sup>	0,25	0,25	0,31	0,04

De acordo com os resultados exibidos na tabela acima, com a utilização dos dados provenientes da estação da Tijuca, o melhor modelo permanece sendo a rede neural GRU, que forneceu os melhores resultados para 4 das 12 variáveis, assim como o modelo ensemble. O resultado aquém do esperado para o ensemble de modelos pode ter ocorrido devido à natureza estocástica das redes neurais. Testes futuros devem ser realizados para comprovar o desempenho dos modelos nesta situação específica.

### 6.2.3 Copacabana

**Tabela 9** comparação do desempenho do modelo ensemble com o desempenho das redes simples, utilizando dados da estação de Copacabana

coluna	métrica	Ensemble	LSTM	GRU	RNN
Temperatura	RMSE	2,99	3,05	3,09	3,07
NOx	RMSE	24,94	25,14	25,22	24,95
CO	RMSE	0,19	0,19	0,20	0,19
Chuva	RMSE	0,85	0,85	0,86	0,86
Pressão	RMSE	2,87	2,89	3,04	2,91
Velocidade do vento	RMSE	0,41	0,41	0,41	0,41
SO <sub>2</sub>	RMSE	4,71	4,79	4,84	4,67
PM10	RMSE	12,53	13,09	12,61	12,69
O <sub>3</sub>	RMSE	12,84	13,12	12,78	13,67
UR	RMSE	6,76	6,94	6,89	7,14
NO	RMSE	16,76	16,95	16,62	16,44
NO <sub>2</sub>	RMSE	14,92	15,36	14,90	15,07
Temperatura	MAE	2,35	2,42	2,43	2,40
NOx	MAE	19,54	19,75	19,90	19,49
CO	MAE	0,16	0,16	0,16	0,16

Chuva	MAE	0,58	0,58	0,59	0,62
Pressão	MAE	2,26	2,29	2,42	2,26
Velocidade do vento	MAE	0,33	0,33	0,33	0,33
SO <sub>2</sub>	MAE	3,84	3,91	3,94	3,81
PM10	MAE	10,13	10,55	10,12	10,12
O <sub>3</sub>	MAE	10,29	10,47	10,23	10,94
UR	MAE	5,35	5,52	5,47	5,70
NO	MAE	13,12	13,41	13,08	12,65
NO <sub>2</sub>	MAE	11,46	11,88	11,49	11,69
Temperatura	R <sup>2</sup>	0,15	0,15	0,15	0,22
NOx	R <sup>2</sup>	0,10	0,06	0,02	0,15
CO	R <sup>2</sup>	-4,08	-4,17	-4,10	-5,36
Chuva	R <sup>2</sup>	-4,60	-4,23	-4,06	-4,94
Pressão	R <sup>2</sup>	0,34	0,32	0,26	0,45
Velocidade do vento	R <sup>2</sup>	-7,53	-5,51	-6,44	-5,85
SO <sub>2</sub>	R <sup>2</sup>	-2,33	-2,06	-2,23	-2,42
PM10	R <sup>2</sup>	-1,08	-1,11	-0,93	-0,77
O <sub>3</sub>	R <sup>2</sup>	-1,30	-1,36	-1,19	-1,33
UR	R <sup>2</sup>	-1,28	-1,19	-1,21	-1,01
NO	R <sup>2</sup>	0,14	0,08	0,08	0,23
NO <sub>2</sub>	R <sup>2</sup>	0,33	0,35	0,30	0,32

Conforme os resultados exibidos na tabela acima, de acordo com as métricas RMSE e MAE, o modelo ensemble foi superior para 8 das 12 variáveis, quando treinado com os dados da estação de Copacabana. O modelo ensemble atingiu também métricas próximas às melhores nas outras 4 estações. Para checar a evolução dos resultados até esta etapa é feita a comparação dos resultados das previsões dos poluentes do modelo ensemble e do modelo inicial, nas variáveis em que o ensemble performou melhor.

Na primeira etapa o melhor modelo obteve RMSE de 25.14 e MAE de 19.81 para NOx, ao passo que o modelo ensemble obteve RMSE de 24.94 e MAE de 19.49, sendo esta uma das variáveis ausentes, preenchidas com o método explicado anteriormente. O primeiro modelo teve RMSE de 0.27 e MAE de 0.22 para a variável CO, o modelo ensemble teve 0.19 e 0.16 para RMSE e MAE, respectivamente, com uma média de 0.15 para esta variável. 3.38

#### 6.2.4 Centro

**Tabela 10** comparação do desempenho do modelo ensemble com o desempenho das redes simples, utilizando dados da estação do Centro

coluna	métrica	Ensemble	LSTM	GRU	RNN
Temperatura	RMSE	3,33	3,38	3,45	3,50
NOx	RMSE	25,01	24,96	25,52	24,74
CO	RMSE	0,22	0,23	0,22	0,23
Chuva	RMSE	0,79	0,79	0,80	0,81

Pressão	RMSE	2,56	2,79	2,94	2,94
Velocidade do vento	RMSE	0,50	0,52	0,51	0,52
SO <sub>2</sub>	RMSE	5,32	5,36	5,48	5,51
PM10	RMSE	14,38	14,62	14,65	14,70
O <sub>3</sub>	RMSE	15,75	16,09	15,86	15,99
UR	RMSE	8,33	8,54	8,21	8,63
NO	RMSE	16,28	16,62	16,36	17,44
NO <sub>2</sub>	RMSE	14,69	14,82	15,12	15,57
Temperatura	MAE	2,64	2,84	2,71	2,73
NOx	MAE	19,45	20,77	20,37	19,62
CO	MAE	0,17	0,19	0,18	0,19
Chuva	MAE	0,54	0,60	0,55	0,55
Pressão	MAE	2,00	2,32	2,32	2,31
Velocidade do vento	MAE	0,41	0,44	0,42	0,42
SO <sub>2</sub>	MAE	4,38	4,61	4,48	4,54
PM10	MAE	11,61	12,41	11,79	11,88
O <sub>3</sub>	MAE	12,72	13,62	12,76	12,95
UR	MAE	6,37	6,61	6,01	6,47
NO	MAE	12,74	13,93	12,97	13,77
NO <sub>2</sub>	MAE	11,36	12,17	11,70	12,11
Temperatura	R <sup>2</sup>	-0,22	-0,18	-0,36	-0,23
NOx	R <sup>2</sup>	0,13	0,15	0,13	0,13
CO	R <sup>2</sup>	-2,08	-1,92	-1,84	-1,93
Chuva	R <sup>2</sup>	-4,44	-4,00	-4,04	-4,76
Pressão	R <sup>2</sup>	0,51	0,39	0,25	0,35
Velocidade do vento	R <sup>2</sup>	-4,63	-4,99	-4,32	-4,45
SO <sub>2</sub>	R <sup>2</sup>	-2,37	-2,08	-2,43	-2,57
PM10	R <sup>2</sup>	-0,79	-0,94	-0,81	-0,78
O <sub>3</sub>	R <sup>2</sup>	-1,07	-1,03	-1,11	-1,20
UR	R <sup>2</sup>	-2,86	-2,75	-2,34	-2,34
NO	R <sup>2</sup>	0,17	0,10	0,17	-0,01
NO <sub>2</sub>	R <sup>2</sup>	0,36	0,39	0,33	0,19

Em acordo com os resultados exibidos na tabela acima, o modelo ensemble treinado com os dados da estação do Centro, foi melhor que os modelos mais simples para 9 das 12 variáveis que compõem o conjunto de dados.

A evolução dos resultados ficou mais evidente quando se compararam os RMSE e MAE obtidos nesta etapa para o modelo ensemble, com as métricas obtidas para o melhor modelo inicial. Para o poluente CO, o modelo inicial obteve RMSE de 0.22 e MAE de 0.18, o modelo ensemble obteve os mesmos índices. Para o poluente PM<sub>10</sub> o melhor modelo inicial obteve RMSE de 14.69 e MAE de 11.85, o modelo ensemble obteve 14.38 e 11.61 para RMSE e MAE respectivamente, com uma média de 33.35 para a variável. O melhor modelo da primeira etapa obteve RMSE de 15.87 e MAE de 12.82 , o modelo ensemble obteve RMSE de 15.75 e MAE de 12.72, com uma média de 33.18 para a variável . Na primeira etapa o melhor modelo obteve RMSE de 14.73 e

MAE de 11.38 para a variável NO<sub>2</sub>, o modelo ensemble obteve RMSE de 14.69 e MAE de 11.36, com uma média de 33.45 para esta variável.

## 7 Conclusão e trabalhos futuros

Este trabalho teve por objetivos: estudar como a variação climática influencia na qualidade atmosférica; examinar a correlação entre os dados de estações atmosféricas adjacentes; utilizar as informações obtidas na exploração de dados na construção de um ensemble de modelos de aprendizado de máquina, com o intuito de prever os níveis máximos diários dos principais poluentes atmosféricos.

Para melhor entendimento dos dados e das metodologias utilizados no trabalho, foram realizadas revisões de bibliografia a respeito das variáveis que constituem o conjunto de dados e da teoria por trás das redes neurais. Foi feita também uma análise exploratória onde foi constatada a presença de dados espúrios e nulos. A metodologia utilizada para o preenchimento dos dados, a substituição dos valores ausentes pela média ponderada dos valores das estações mais próximas, foi escolhida com base na boa correlação entre os dados de estações adjacentes.

Com o objetivo de definir a melhor topografia das redes neurais estudadas bem como o melhor conjunto de hiperparâmetros das mesmas, foram realizados testes preliminares que indicaram que redes de uma camada intermediária, com 64 neurônios e com tamanhos de *batches* menores, tendem a fornecer melhores resultados. Os resultados indicaram também que, para o problema estudado, a rede neural simples que fornece os melhores resultados é a GRU. Ficou também constatado que um número maior que 150 épocas de treinamento não melhora a qualidade das previsões, podendo inclusive piorá-los em determinados casos. Não houve melhora nos resultados fornecidos ao se empregarem normalização de *batch* ou *dropout* as redes. Todavia um *dropout* de 20% acelerou o treinamento sem perdas consideráveis de desempenho. Por fim, o *ensemble* de redes neurais forneceu melhores resultados para três das quatro estações estudadas, tendo tido um desempenho similar ao da rede GRU em apenas uma das estações.

Desta forma, de acordo com as métricas utilizadas para avaliar os modelos, RMSE, MAE e tempo total de treinamento, é possível considerar que houve uma melhora no desempenho das previsões da poluição atmosférica, quando se compara o modelo *ensemble* com os modelos de rede recorrente padrão (RNN).

É esperado que com estes resultados, em estudos posteriores, a tarefa de elaboração de um modelo ideal, possivelmente utilizando *deep learning*, tenha se tornado mais facilitada. Adicionalmente, com a pesquisa feita na revisão bibliográfica, foram encontradas alternativas mais baratas e/ou eficientes, que podem complementar a rede de coleta de dados de qualidade do ar atual. Por fim, foi sugerida uma arquitetura de computação em nuvem, para o sistema de alerta de níveis de poluentes acima do aceitável, que também poderá servir de aprimoramento para o sistema utilizado atualmente pela prefeitura do Rio de Janeiro.

Uma das dificuldades encontradas na elaboração deste trabalho foi o número de registros com valores nulos, maior que 70% para alguns casos. Uma vez que o método de preenchimento de valores ausentes utilizado neste trabalho se baseia na proximidade geográfica, por não haverem estações próximas, não foram feitas previsões para 4 das 8 estações do conjunto de dados utilizado. Os  $R^2$  dos treinamentos dos modelos supostamente mostram que há um subajuste dos modelos aos dados do problema, todavia segundo alguns autores essa métrica pode não ser adequada para modelos não-lineares. A última dificuldade encontrada se dá pelo fato de não ter sido possível elaborar modelo de aprendizado profundo para o problema. Estes modelos, que supostamente possuem um desempenho superior aos modelos com menos camadas, podem necessitar de um treinamento de maior duração para que o ajuste aos dados seja adequado.

Sugere-se que nos trabalhos subsequentes a este, o foco se dê nas seguintes melhorias:

- Estudar outros modelos de aprendizado de máquina, adequados a séries temporais. Com intuito de incorporá-los ao ensemble, para que o desempenho do modelo atual seja aprimorado;
- Incorporar dados auxiliares ao conjunto de dados utilizado no trabalho, como a informação de trânsito de veículos e localização geográfica de indústrias, para tornar as previsões fornecidas pelo modelo mais assertivas;
- Enriquecer a base de dados, utilizando dados de outras localidades no treinamento dos modelos, com o objetivo de obter um modelo mais

generalista, que possa ser empregado em outros contextos, não somente para os dados de qualidade do ar do Rio de Janeiro;

- Utilizar a técnica de transferência de aprendizagem, para que seja possível o reaproveitamento de modelos previamente treinados com dados de outras estações, facilitando e tornando mais rápido o processo de treinamento do modelo;
- Testar outras janelas de previsão, para estudar como o modelo se comporta ao prever múltiplos dias adiante;
- Experimentar outras metodologias para o *ensemble* de modelos no problema estudado, para aprimorar os resultados já obtidos.

## 8 Referências Bibliográficas

- [1] OMS (Organização Mundial da Saúde). *Burden of Disease from Household Air Pollution for 2016*. 2018.
- [2] O. A. Ghoneim; Doreswamy; B. R. Manjunatha. “Forecasting of Ozone Concentration in Smart City using Deep Learning”. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2017
- [3] A.S. Luna; M. L. L. Paredes; G. C. G. de Oliveira; S. M. Corrêa. “Prediction of ozone concentration in tropospheric levels using artificial neural networks and support vector machine at Rio de Janeiro, Brazil”. *Atmospheric Environment* 98. 2014
- [4] X. Li; L. Peng; Y. Hu; J. Shao; T. Chi. “Deep learning architecture for air quality predictions”. *Environ Sci Pollut Res*.2016.
- [5] I. Kök; M. U. Simsek; S. Özdemir. “A deep learning model for air quality prediction in smart cities”. *IEEE International Conference on Big Data (BIGDATA)*, 2017.
- [6] J. Fan; Q. Li; J. Hou; X.Feng; H. Karimian; S. Lin. “A Spatiotemporal Prediction Framework for Air Pollution Based on Deep RNN”. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume IV-4/W2*. 2017.
- [7] X. Li; L. Peng; X. Yao; S. Cui; Y. Hu; C. You. “Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation”. *Environmental Pollution* 231. 2017.
- [8] T. Bui; V. Le; S. K.Cha. “A Deep Learning Approach for Forecasting Air Pollution in South Korea Using LSTM”. *Environmental Pollution* 231. 2018.
- [9] H. Sak; G. Yang; B. Li; W. Li. “Modeling Dependence Dynamics of Air Pollution: Pollution Risk Simulation and Prediction of PM2.5 Levels”. 2016.
- [10] E. Pardo; N. Malpica. “Air Quality Forecasting in Madrid Using Long Short-Term Memory Networks”. *IWINAC*. 2017.
- [11] J. Wang; X. Zhang; Z. Guo; H. Lu. “Developing an early-warning system for air quality prediction and assessment of cities in China”. *Expert Systems With Applications*. 2017.
- [12] R. Navares; J. L. “Aznarte. Predicting air quality with deep learning LSTM: Towards comprehensive models”. *Ecological Informatics*.2019.
- [13] E. Hossain; M. A. U. Shariff; M. S. Hossain; K. Andersson. “A Novel Deep Learning Approach to Predict Air Quality Index”. *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*. 2020.

- [14] S. Du: T. Li: Y. Yang: S. J. Horng. “Deep Air Quality Forecasting Using Hybrid Deep Learning Framework”. *IEEE Transactions on Knowledge and Data Engineering*. 2019.
- [15] Q. Zhang: V. O. Li: J. C. Lam: Y. Han. “Deep-AIR: A Hybrid CNN-LSTM Framework for Fine-Grained Air Pollution Forecast”. 2020.
- [16] E. Grossi: M. Buscema. “Introduction to artificial neural networks”. *European Journal of Gastroenterology & Hepatology*, 2008.
- [17] T. Fontes: L. M. Silva: M.P. Silva: N. Barros: A.C. Carvalho. “Can artificial neural networks be used to predict the origin of ozone episodes?”. *Science of the Total Environment*, 2014.
- [18] I. Kök: M. Simsek: S. Ozdemir. “A deep learning model for air quality prediction in smart cities”. *IEEE International Conference on Big Data (BIGDATA)*, 2017.
- [19] X. Li: L. Peng: X. Yao: S. Cui: Y. Hu: C. You. “Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation”. *Environmental Pollution*, 2017.
- [20] E. Pardo: N. Malpica. “Air Quality Forecasting in Madrid Using Long Short-Term Memory Networks”. *Natural and Artificial Computation*, 2017.
- [21]<https://www.deeplearningbook.com.br/arquitetura-de-redes-neurais-long-short-term-memory/>. Acessado em 09/03/2023.
- [22] A. Graves: J. Schmidhuber. “Framewise Phoneme Classification with Bidirectional LSTM Networks”. *IDSIA*. 2005.
- [23]<https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0>. Acessado em 09/03/2023.
- [24] J. Chung: C. Gulcehre: K. Cho: Y. Bengio. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”. 2014.
- [25]<https://blog.floydhub.com/gru-with-pytorch/>. Acessado em 09/03/2023.
- [26]<https://web.archive.org/web/20041108132234/http://www.census.gov/cgi-bin/geo/gisfaq?Q5.1>. Acessado em 09/03/2023.
- [27] M. Shanker: M. Y. Hu: M. S. Hung. “Effect of data standardization on neural network training”. *The International Journal of Management Science*. 1996.
- [28] R. Navares: J. L. Aznarte. “Predicting air quality with deep learning LSTM: Towards comprehensive models”. *An International Journal on Ecoinformatics and Computational Ecology*. 2019.

- [29] E. Hossain: M. Shariff: M. Hossain: K. Anderson. “A Novel Deep Learning Approach to Predict Air Quality Index”. *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*. 2021.
- [30] J. Li: T. Chow: Y. Yu. “The estimation theory and optimization algorithm for number of hidden units in the higher-order feedforward neural network”. *Proceedings of ICNN’95- International Conference on Neural Networks*. 1995.
- [31] N. Petersen: F. Rodrigues: F. Pereira. “Multi-output Bus Travel Time Prediction with Convolutional LSTM Neural Network”. *Arxiv*. 2019.
- [32][https://www.researchgate.net/figure/Classical-k-fold-cross-validation-vs-time-series-split-cross-validation\\_fig1\\_355889701](https://www.researchgate.net/figure/Classical-k-fold-cross-validation-vs-time-series-split-cross-validation_fig1_355889701). Acessado em 09/03/2023.
- [33] V. Cerqueira: L. Torgo: I. Mozetic. “Evaluating time series forecasting models”. *Arxiv*. 2019.
- [34] M. Schnaubelt. “A comparison of machine learning model validation schemes for non-stationary time series data”. *FAU Discussion Papers in Economics*. 2019.
- [35] D. Kingma: J. Ba. “Adam: A Method for Stochastic Optimization”. *Arxiv*. 2014.
- [36] N. Srivastava: G. Hinton: A. Krizhevsky, I. Sutskever. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. *Journal of Machine Learning Research*. 2014.
- [37] J. Whang: A. Matsukawa. “Exploring *Batch* Normalization in Recurrent Neural Networks”. *Universidade Stamford*.
- [38] A. Krizhevsky: I. Sutskever: G. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. *Advances in Neural Information Processing Systems* 25. 2012.
- [39] C. Bishop. “Neural Networks for Pattern Recognition”. *Editora Oxford University Press*. 1996. 367 p.
- [40] AoT (Array of Things). 2020. Chicago, Estados Unidos. Universidade de Chicago. Disponível em: <https://arrayofthings.github.io/>. Acesso em 18/08/2022.
- [41] IBM. 2016. Estados Unidos. Disponível em: <https://www.ibm.com/blogs/internet-of-things/air-pollution-green-initiatives/> Acesso em: 18/08/2022.
- [42] Earthsense. 2020. Inglaterra. Disponível em: <https://www.earthsense.co.uk/> Acesso em: 18/08/2022.
- [43] Google Earth Solidário. 2020. Califórnia, Estados Unidos: Google. Disponível em: <https://www.google.com/earth/outreach/special-projects/air-quality/> Acesso em 18/08/2022.

- [44] X. Liu: B. Li: A. Jiang : S. Qi: C. Xiang: N. Xu. “A Bicycle-borne Sensor for Monitoring Air Pollution near Roadways”. *2015 International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, 2015.
- [45] H. Haddad : A. De Nazelle: “The role of personal air pollution sensors and smartphone technology in changing travel behaviour”. *Journal of Transport & Health*, 2018.
- [46] J. Wang: X. Zhang: Z. Guo: H. Lu. “Developing an early-warning system for air quality prediction and assessment of cities in China”. *Expert Systems With Applications*, 2017
- [47] A. Luna: M. Paredes: C. de Oliveira: S. Corrêa, S. “Prediction of ozone concentration in tropospheric levels using artificial neural networks and support vector machine at Rio de Janeiro”. *Atmospheric Environment*, 2014.
- [48] IPP (Instituto Pereira Passos). 2022. Rio de Janeiro. Prefeitura do Rio de Janeiro. Disponível em: <http://www.rio.rj.gov.br/web/ipp>. Acesso em: 05/10/2022.
- [49] MonitorAr – Rio (Programa de Monitoramento da Qualidade do Ar). 2022. Rio de Janeiro. Secretaria Municipal do Meio Ambiente. Disponível em: <https://www.rio.rj.gov.br/web/smac/monitorar-rio1>. Acesso em: 05/10/2022.
- [50]<http://www.rio.rj.gov.br/dlstatic/10112/3252594/4114836/RelatorioMonitorar20112012.pdf>. Acessado em 09/03/2023.
- [51] CONAMA (Conselho Nacional do Meio Ambiente). 2020. Brasil. Ministério do Meio Ambiente. Disponível em: <http://www2.mma.gov.br/port/conama/> . Acesso em 05/10/2022.
- [52] CONAMA (Conselho Nacional do Meio Ambiente). 2020. Brasil. Ministério do Meio Ambiente. Disponível em: <http://www2.mma.gov.br/port/conama/> . Acesso em 05/10/2022.
- [53] M. Muladi: S. Sendari: T. Widiyaningtyas. “Outdoor Ar Quality Monitor Using MQTT Protocol on Smart Campus Network”. *International Conference on Sustainable Information Engineering and Technology (SIET)*. 2018.
- [54] W. W. Daniel.”Spearman rank correlation coefficient”. *Applied Nonparametric Statistics (2nd ed.)*.1990.
- [55] F. Chollet. “Deep Learning with Python”. *Editores Manning*. 2017. 72 p.
- [56] Spiess: A. Nikolai: N. Neumeyer. “An evaluation of  $R^2$  as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach”. *BMC Pharmacology*. 2010.

- [57] S. Namini: N. Tavakoli: A. Namin. “The Performanc of LSTM and BiLSTM in Forecasting Time Series”. *2019 IEEE International Confernce on Big Data (Big Data)*. 2019.
- [58] J. Chung: C. Gulcehre: K. Cho: Y. Bengio. “Empirical Evaluation of Gated Recurrent Nural Networkson Sequence Modeling”. *Arxiv*. 2014.
- [59] S. Yang: X. Yu: Y. Zhou. “LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as na Example”. *2020 International Workshop on Eletronic Communication and Artificial Intelligende (IWECAI)*. 2020.
- [60] R. Pascanu: C. Gulcehre: K. Cho: Y. Bengio. “How to Construct Deep Recurrent Neural Networks”. *Arxiv*. 2013.
- [61] M. Hermans: B. Schrauwen. “Training and Analysing Deep Recurrent Neural Networks”. *Advances in Neural Information Processing Systems* 26. 2013.
- [62] N. Srivastava: G. Hinton: A. Krizhevsky: I. Sutskever. “Dropout: A Simple Way to Prevent Networks from Overfitting”. *Journal of Machine Learning Research*. 2014.
- [63] S. Ioffe: C. Szegedy. “Bach Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. *Proceedings of the 32nd International Conference on Machine Learning*. 2015.
- [64] T. Coojimans: N. Ballas: C. Laurent: Ç. Güçehre: A. Courville. “Recurrent *Batch* Normalization”. *Arxiv*. 2016.
- [65] J. Whang: A. Matsukawa. “Exploring *Batch* Normalization in Recurrent Neural Networks”. *Universidade Stamford*.
- [66] D. Huynh. “Applying Dropout to Prevent Shallow Neural Networks from Overtraining”. *Universidade de Lund*. 2017.