CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

INTERPOLAÇÃO DE PRECIPITAÇÃO POR REDES NEURAIS CONVOLUCIONAIS PARA GRAFOS

Augusto José Moreira da Fonseca

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de mestre.

Orientadores:
Eduardo Bezerra da Silva, D.Sc.
Fabio Andre Machado Porto, D.Sc.

Rio de Janeiro,

Dezembro de 2024

Interpolação de Precipitação por Redes Neurais Convolucionais para Grafos

Dissertação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Augusto José Moreira da Fonseca

Aprovada por:

Presidente, Eduardo Bezerra da Silva, D.Sc. (CEFET/RJ)

Fabio Andre Machado Porto, D.Sc. (LNCC)

Eduardo Soares Ogasawara, D.Sc. (CEFET/RJ)

Mariza Ferro, D.Sc. (IC/UFF)

Leonardo Silva de Lima, D.Sc. (PPGM/UFPR)

Rio de Janeiro,

Dezembro de 2024

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

F676 Fonseca, Augusto José Moreira da
Interpolação de precipitação por redes neurais convolucionais para grafos / Augusto José Moreira da Fonseca. — 2024.
73f. + apêndice : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2024.
Bibliografia : f. 69-73
Orientador: Eduardo Bezerra da Silva
Coorientador: Fabio Andre Machado Porto

1. Meteorologia. 2. Mudanças climáticas - Previsão. 3. Aprendizado do computador. 4. Teoria dos grafos. 5. Interpolação – Processamento de dados I. Silva, Eduardo Bezerra da. (Orient.). II. Porto, Fabio Andre Machado (Coorient.). III. Título.

CDD 551.5

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

RESUMO

Interpolação de Precipitação por Redes Neurais Convolucionais para Grafos

Augusto José Moreira da Fonseca

Orientadores:
Eduardo Bezerra da Silva, D.Sc.
Fabio Andre Machado Porto, D.Sc.

O monitoramento e a previsão das condições climáticas atmosféricas têm implicações significativas em diversos campos, como agricultura, transporte e segurança pública. Previsões eficazes das condições climáticas podem ajudar a mitigar e prevenir os impactos adversos de eventos climáticos severos, incluindo a perda de vidas humanas. Com os avanços tecnológicos, existem vários instrumentos disponíveis para observação atmosférica, como satélites, radares e estações de superfície. Esses instrumentos fornecem observações em tempo real e, em alguns casos com alta resolução espacial e temporal, tornado-se insumo ideal para o treinamento de modelos de previsão do tempo baseados em Aprendizado de Máquina (AM). Por outro lado, um desafio acerca dos dados meteorológicos é a disponibilidade e distribuição espacial das estações de superfície. Por conta de sua distribuição desigual, certas áreas acabam com monitoramento insuficiente. Métodos de interpolação são comumente usados para inferir valores de precipitação para áreas não cobertas. No entanto, esses métodos frequentemente negligenciam o contexto e a física da atmosfera, resultando em discrepâncias entre os dados interpolados e os observados. Além disso, esses métodos não levam em consideração o espaço e o tempo simultaneamente. Este estudo tem como objetivo implementar uma Rede Convolucional Espaço-Temporal de Grafos (STGCN, na sigla em inglês) para interpolar dados de precipitação, que é uma variável significativamente desbalanceada. A STGCN é adaptada para realizar a interpolação da precipitação, aprendendo os padrões e a física da atmosfera em séries temporais multivariadas de múltiplos instrumentos meteorológicos. O objetivo é melhorar a precisão da interpolação, especialmente em áreas com cobertura limitada de estações de superfície. Comparamos os resultados alcançados pela STGCN com os de um método tradicional de interpolação, o Inverse Distance Weighting (IDW). Os resultados alcançados apontam que, comparado ao IDW, a STGCN é capaz de obter melhor acurácia em eventos mais extremos de precipitação e alcançam resultados promissores em eventos de menor magnitude. Em alguns casos, nosso modelo chegou a alcançar uma acurácia para eventos mais extremos entre 40% e 80% maior que o método IDW. No entanto, identificamos que o desbalanceamento dos dados de precipitação e a relativa baixa quantidade de exemplos de eventos extremos podem prejudicar a acurácia da STGCN em alguns cenários. Nosso método se mostra promissor e abre caminho para futuras pesquisas.

Palavras-chave:
Meteorologia, Previsão de curto prazo, Aprendizado de Máquina, Redes Convolucionais para Grafos, Interpolação de Dados.

Rio de Janeiro,

Dezembro de 2024

ABSTRACT

Interpolação de Precipitação por Redes Neurais Convolucionais para Grafos

Augusto José Moreira da Fonseca

Advisors:
Eduardo Bezerra da Silva, D.Sc.
Fabio Andre Machado Porto, D.Sc.

The monitoring and forecasting of atmospheric weather conditions have significant implications across diverse fields, including agriculture, transportation, and public safety. Accurate weather forecasting can help mitigate and prevent the adverse impacts of severe weather events, including the loss of human lives. With technological advancements, several instruments are available for atmospheric observation, such as satellites, radars, and in-situ stations. These instruments provide real-time observations and, in some cases, high spatial and temporal resolution, making them ideal input for training Machine Learning (ML)-based weather prediction models. However, one challenge regarding meteorological data is the availability and spatial distribution of in-situ stations. Due to their uneven distribution, certain areas end up with insufficient monitoring. Interpolation methods are commonly used to infer precipitation values for areas not covered by in-situ stations. However, these methods often overlook atmospheric context and physics, leading to discrepancies between interpolated and observed data. Additionally, these methods do not simultaneously account for both spatial and temporal dimensions. This study aims to implement a Spatio-Temporal Graph Convolutional Network (STGCN) to interpolate precipitation data, a significantly imbalanced variable. The STGCN is adapted to perform precipitation interpolation by learning atmospheric patterns and physics in multivariate time series from multiple meteorological instruments. The objective is to improve interpolation accuracy, especially in areas with limited in-situ station coverage. We compare the results achieved by the STGCN with those of a traditional interpolation method, Inverse Distance Weighting (IDW). The results indicate that, compared to IDW, the STGCN achieves higher accuracy in extreme precipitation events and promising results in events of lower magnitude. In some cases, our model achieved accuracy improvements for extreme events ranging from 40% to 80% over the IDW method. However, we identified that the imbalance in precipitation data and the relatively low number of extreme event examples could hinder STGCN accuracy in some scenarios. Our method proves promising and paves the way for future research.

Key-words:

Meteorology, Nowcasting, Machine Learning, Graph Convolutional Neural Netwworks, Data Interpolation.

Rio de Janeiro,

Dezembro de 2024

Sumário

I Introdução **1**

I.1 Contextualização 1

I.2 Motivação 1

I.3 Objetivos 4

I.4 Metodologia 5

I.5 Organização 6

II Referencial Teórico **7**

II.1 Radar Meteorológico 7

II.2 Estações Meteorológicas 9

II.3 Interpolação de Dados Geolocalizados 10

II.4 Grafos 11

II.5 Artificial Neural Network 12

II.5.1 Perceptron 12

II.5.2 Multilayer Perceptron 14

II.6 Graph Convolutional Network 16

II.7 Séries Temporais 17

II.8 Gated Recurrent Unit 19

II.9 Attention Mechanism 20

II.10 Spatiotemporal Graph Convolutional Network 21

III Trabalhos Relacionados **23**

IV Metodologia **27**

IV.1 Formulação do Problema 27

IV.2 Fontes de Dados 29

IV.3 Data Lake 29

IV.4 Arquitetura STGCN 32

V Avaliação Experimental **34**

V.1 Análise Exploratória dos Dados 34
V.1.1 Radar Meteorológico do INEA 35
V.1.2 Estações Meteorológicas do INMET 38
V.1.3 Estações Meteorológicas do Alerta Rio 40
V.1.4 Estações Pluviométricas do Alerta Rio 41
V.2 Construção do Grafo 43
V.3 Pré-processamento dos Dados 46
V.3.1 Pré-processamento de Dados de Estações Telemétricas 47
V.3.2 Pré-processamento de Dados de Radar 48
V.3.3 Integração dos Dados e Construção das Séries Temporais de Grafos 48
V.4 Treinamento e Avaliação do Modelo 50

VI Resultados e Análise **51**
VI.1 Resultados 51
VI.1.1 Estação Tijuca Muda 52
VI.1.2 Estação Guaratiba 53
VI.1.3 Estação Alto da Boa Vista 55
VI.1.4 Demais Estações 56
VI.2 Análise Retrospectiva 59

VII Conclusão e Trabalhos Futuros **66**
VII.1 Conclusão 66
VII.2 Trabalhos Futuros 68

Referências **69**

A Estações Vizinhas **74**

Lista de Figuras

I.1 Localização das estações telemétricas do RJ 3

I.2 Comparação entre valores registrados e interpolados 4

II.1 Escaneamento por radar meteorológico 8

II.2 Feixe de energia do radar meteorológico 9

II.3 Cone do Silêncio 10

II.4 Matriz de Adjacência 12

II.5 Perceptron 13

II.6 Multilayer Perceptron 14

IV.1 Visão geral da arquitetura proposta 28

IV.2 Localização dos instrumentos de observação 30

IV.3 *Data Lake* 31

IV.4 Visão geral da STGCN 32

V.1 Estratégia de escaneamento do radar do INEA 35

V.2 Análise de operação do radar 37

V.3 Cone do Silêncio na área de estudo 37

V.4 Percentual de valores nulos das estações do INMET 39

V.5 Distribuição da variável dew_point - INMET 39

V.6 Percentual de valores nulos das estações meteorológicas do Alerta Rio 41

V.7 Distribuição da variável *wind_speed* - Alerta Rio 42

V.8 Distribuição da variável *temperature* - Alerta Rio 43

V.9 Percentual de valores nulos das estações pluviométricas do Alerta Rio 44

V.10 Histograma da variável *precipitation* 44

V.11 Grade regular $\mathbf{\Gamma}$ de pontos 45

V.12 Grafo da grade irregular $\mathbf{\Upsilon}$ de pontos 45

VI.1 Tijuca Muda - Análise de Influência da Vizinhança 52

VI.2 Curvas de Aprendizado - Tijuca Muda 53

VI.3 Guaratiba - Análise de Influência da Vizinhança 54
VI.4 Curvas de Aprendizado - Guaratiba 54
VI.5 Alto da Boa Vista - Análise de Influência da Vizinhança 55
VI.6 Curvas de Aprendizado - Guaratiba 56
VI.7 Campo Grande - Análise de Influência da Vizinhança 57
VI.8 Grota Funda - Análise de Influência da Vizinhança 57
VI.9 Riocentro - Análise de Influência da Vizinhança 58
VI.10Curvas de Aprendizado - Campo Grande 58
VI.11Curvas de Aprendizado - Grota Funda 58
VI.12Curvas de Aprendizado - Riocentro 59
VI.13Histograma Tijuca Muda x Campo Grande 60
VI.14Histograma Tijuca Muda x Grota Funda($\mathcal{N}_6$) 61
VI.15Histograma Tijuca Muda x Grota Funda($\mathcal{N}_3$) 61
VI.16Histograma Tijuca Muda x Riocentro 61
VI.17Histograma Tijuca Muda x Alto da Boa Vista($\mathcal{N}_6$) 62
VI.18Topografia da Região de Alto da Boa Vista 62
VI.19Histograma Tijuca Muda x Alto da Boa Vista($\mathcal{N}_3$) 63
VI.20Histograma Tijuca Muda x Guaratiba 63
VI.21Correlação entre Refletividade e Precipitação 64
VI.22Correlação entre Refletividade e Precipitação por Ano 65

Lista de Tabelas

II.1 Exemplos criados pela técnica de janelamento 19

III.1 Visão geral dos trabalhos relacionados 26

V.1 Classificação dos níveis de precipitação 35
V.2 Coordenadas e período de operação das estações do INMET 38
V.3 Seleção das variáveis observadas pelas estações do INMET 38
V.4 Coordenadas e período de operação das estações meteorológicas do Alerta Rio 40
V.5 Variáveis observadas pelas estações meteorológicas do Alerta Rio 40
V.6 Coordenadas das estações pluviométricas do Alerta Rio 42
V.7 Valores limites das variáveis meteorológicas 46
V.8 Estações de Interesse 47
V.9 Exemplo de Integração de Dados - Tabela A 49
V.10 Exemplo de Integração de Dados - Tabela B 49
V.11 Exemplo de Integração de Dados - Tabela C 49
V.12 Exemplo de Integração de Dados - Tabela D 49

VI.1 Tijuca Muda - RMSE 53
VI.2 Guaratiba - RMSE 55
VI.3 Alto da Boa Vista - RMSE 56

A.1 Tijuca Muda - Estações Vizinhas 74
A.2 Guaratiba - Estações Vizinhas 74
A.3 Alto da Boa Vista - Estações Vizinhas 75
A.4 Campo Grande - Estações Vizinhas 75
A.5 Grota Funda - Estações Vizinhas 75
A.6 Riocentro - Estações Vizinhas 76

Lista de Abreviações

AM	Aprendizado De Máquina	1, 2, 3, 31, 43, 46
ANN	*Artificial Neural Network*	12, 16, 17, 18, 19, 21, 24, 26, 28
CEFET/RJ	Centro Federal De Educação Tecnológica Celso Suckow Da Fonseca	2
CHC	*Climate Hazards Center*	2, 3, 4, 25
CNN	*Convolutional Neural Network*	25
ETL	*Extract Transform Load*	5
FFNN	*Feed Forward Neural Network*	15
GCN	*Graph Convolutional Network*	16, 17, 21, 32
GPCC	*Global Precipitation Climatology Centre*	2, 3, 25
GRU	*Gated Recurrent Unit*	19, 32
IBGE	Instituto Brasileiro De Geografia E Estatística	1
IDW	*Inverse Distance Weighting*	11, 50, 51, 52, 53, 54, 55, 56, 59, 63, 66
INEA	Instituto Estadual Do Ambiente	5, 29, 30, 34, 35, 36, 40, 41, 42
INMET	Instituto Nacional De Meteorologia	5, 29, 30, 34, 47, 48
LNCC	Laboratório Nacional De Computação Científica	2
MLP	*Multilayer Perceptron*	13, 14, 16, 32, 33, 50
NWP	*Numerical Weather Prediction*	2
RELU	*Rectified Linear Unit*	13
RNN	*Recurrent Neural Network*	19, 21
STCONVS2S	*Spatiotemporal Convolutional Sequence to Sequence Network*	25
STGCN	*Spatiotemporal Graph Convolutional Network*	21, 22, 23, 24, 25, 26, 27, 28, 32, 34, 45, 50, 51, 52, 53, 54, 55, 56, 57, 59, 60, 66, 68
UFF	Universidade Federal Fluminense	2

Capítulo I Introdução

I.1 Contextualização

A relevância do monitoramento e previsão das condições climáticas da atmosfera justifica-se em várias áreas como por exemplo agricultura, transporte, segurança, dentre outras. As consequências dos diversos tipos de eventos climáticos podem assumir escalas extremas como a perda de um grande número de vidas humanas [Lu et al., 2020; Azadi et al., 2022; Hoeppe, 2016]. Tais impactos ao meio ambiente e à sociedade podem ser mitigados ou até mesmo prevenidos por meio do monitoramento e previsão eficazes dessas condições climáticas.

O monitoramento das condições atmosféricas vem ganhando cada vez mais tipos de instrumentos de observação devido aos avanços tecnológicos. Dentre os instrumentos, podemos citar os satélites, radares, estações telemétricas, boias marítimas e radiossondas (balões meteorológicos). Dependendo do instrumento e tecnologia associada, observações são registradas e disponibilizadas em poucos segundos [Goodman et al., 2013], gerando um grande volume de dados. Meteorologistas monitoram e analisam os dados recentes observados por tais instrumentos e, a partir de suas expertises, estimam as condições climáticas futuras, processo este popularmente conhecido como *previsão do tempo*.

Por outro lado, tal volume de dados converte-se em um cenário ideal para o treinamento de modelos de previsão do tempo baseados em Aprendizado de Máquina (AM) [Ren et al., 2021a]. Arquiteturas de AM podem ser modeladas para receber como entrada os valores observados pelos diversos instrumentos meteorológicos e gerar como saída a previsão das condições climáticas futuras. Estudos recentes [Castro et al., 2021; Lewenfus et al., 2020; Tekin et al., 2021] vêm alcançando resultados preliminares, demostrando certos níveis de eficácia desses modelos na captura e aprendizado dos padrões espaciais e temporais dos dados meteorológicos.

I.2 Motivação

A cidade do Rio de Janeiro está localizada na região sudeste do Brasil, possui uma área territorial aproximada de 1.200 km^2 e sua população estimada em 2021 era de mais de 6,7 milhões de habitantes [IBGE, 2023]. De acordo com o censo demográfico do Instituto Brasileiro de Geografia e Estatística (IBGE) de 2010, a estimativa de população vivendo em áreas de risco de desastres naturais era de 7% [IBGE, 2018]. Desastres como deslizamentos em encostas, inundações e alagamentos

são causados pelas chuvas intensas na cidade. A topografia, extensão litorânea e proximidade com o Oceano Atlântico são alguns dos fatores que contribuem para uma grande variabilidade espacial e temporal da precipitação na cidade [Dereczynski et al., 2009]. Diante deste cenário, modelos de previsão tornam-se elementos fundamentais na prevenção e mitigação dos impactos sociais causados pelas chuvas intensas.

Em 2021 foi estabelecido um consórcio[1] entre a Prefeitura da Cidade do Rio de Janeiro e as instituições Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ) e Laboratório Nacional de Computação Científica (LNCC), formando uma parceria para aprimorar a previsão do tempo na cidade do Rio de Janeiro a partir de pesquisas baseadas em Inteligência Artificial e AM. A iniciativa conduziu estudos técnicos e troca de informações entre as instituições federais e órgãos municipais, como o Alerta Rio [AlertaRio, 2023]. Mais tarde, o consórcio idealizou o projeto *RioNowcast* [2], que tem por objetivo específico desenvolver modelos baseados em AM para previsão de eventos extremos de precipitação no Rio de Janeiro. Atualmente, também faz parte do consórcio a Universidade Federal Fluminense (UFF).

O Alerta Rio é um serviço de monitoramento, previsão e sistema de alerta de possibilidades de chuvas intensas, deslizamentos, inundações, dentre outros acidentes geotécnicos. O sistema Alerta Rio conta com uma rede de 33 estações telemétricas (25 pluviométricas, 6 estações meteorológicas parciais e 2 estações meteorológicas completas). O Alerta Rio, por meio desses dados e de sua equipe de meteorologistas, emprega modelos de *previsão numérica do tempo*, do inglês *Numerical Weather Prediction* (NWP) [Kimura, 2002] para estimar as condições climáticas futuras e gerar boletins de alerta à população.

Um problema inerente ao emprego dos dados de estações telemétricas na previsão do tempo consiste na distribuição não homogênea e baixa densidade espacial em relação à área considerada [Lyra et al., 2018]. Como efeito destes dois fatores, algumas subáreas podem não ser devidamente monitoradas. A Figura I.1 apresenta a localização das estações telemétricas da cidade do Rio de Janeiro, em que observa-se os dois fatores mencionados. Pontos em azul denotam a localização das estações telemétricas, e pontos em vermelho os limites da área territorial de estudo deste trabalho.

Métodos de interpolação são empregados para inferir valores de precipitação para áreas não cobertas por estações telemétrica, como forma de contornar o problema supracitado. Por exemplo, o *Global Precipitation Climatology Centre* (GPCC) [Liebmann and Allured, 2005] aplica métodos de interpolação a dados de estações meteorológicas na construção de conjuntos de dados de precipitação em forma de grade (latitude x longitude), com resolução espacial de 1 grau. Em outra abordagem, o *Climate Hazards Center* (CHC) [Funk et al., 2015] combina dados de estações telemétricas e de

[1] https://www.gov.br/lncc/pt-br/assuntos/noticias/ultimas-noticias-1/cor-lncc-e-cefet-rj-fazem-parceria-para-aprimorar-a-previsao-do-tempo-na-capital-fluminense

[2] https://rionowcast.dexl.lncc.br/

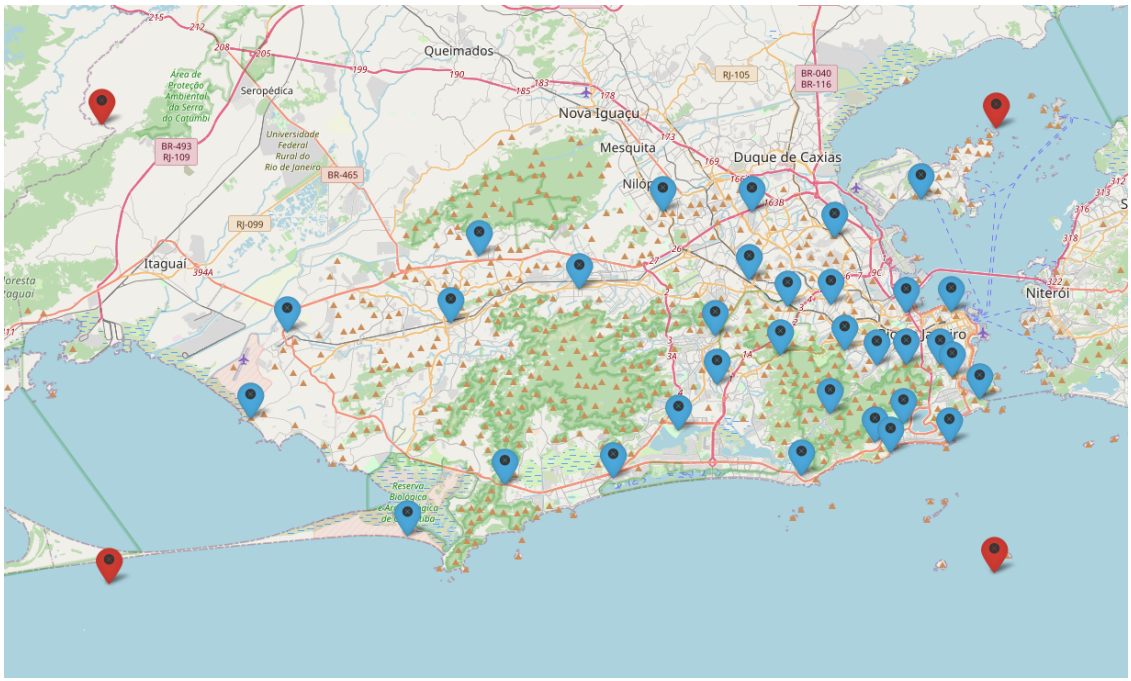


Figura I.1: Localização das estações telemétricas da cidade do Rio de Janeiro. Pontos em azul denotam as estações telemétricas. Pontos em vermelho denotam os limites da área territorial de estudo.

satélites no processo de interpolação. Os conjuntos de dados resultantes também são gerados em forma de grade, com resolução espacial de 0.05 graus.

A desvantagem desses métodos reside na ocorrência de valores discrepantes entre os dados interpolados e os dados registrados pelas estações telemétricas [Lyra et al., 2018]. De fato, a Figura I.2 demonstra tal discrepância entre o valor registrado por uma estação telemétrica da cidade do Rio de Janeiro e o valor interpolado do conjunto de dados do CHC para os primeiros 14 dias do mês de janeiro de 2020. Como pode ser observado na Figura I.2, o valor interpolado pelo CHC tende a superestimar os valores de precipitação no período apresentado.

O treinamento de um modelo de AM é sensível à qualidade dos sinais que são dados como entrada para o algoritmo de aprendizado. Dados imprecisos introduzem erro no treinamento de um modelo o que provavelmente diminuirá sua acurácia. Tal imprecisão pode ter origem nas discrepâncias processadas pela interpolação dos dados, uma vez que técnicas comuns de interpolação não exploram bem o contexto e a física da atmosfera.

Trabalhos que empregam AM no treinamento de modelos de previsão do tempo usam, em geral, três tipos de abordagem em relação aos conjuntos de dados: *i)* empregam conjuntos de dados já interpolados similares aos disponibilizados pelo GPCC e CHC; *ii)* aplicam sua própria estratégia de interpolação aos dados de estações telemétricas para transformá-los em um conjunto de dados em forma de grade e; *iii)* empregam dados de estações telemétricas sem aplicar interpolação, o que limita o modelo de previsão às áreas cobertas pelas estações [Lewenfus et al., 2020; Zhou et al.,

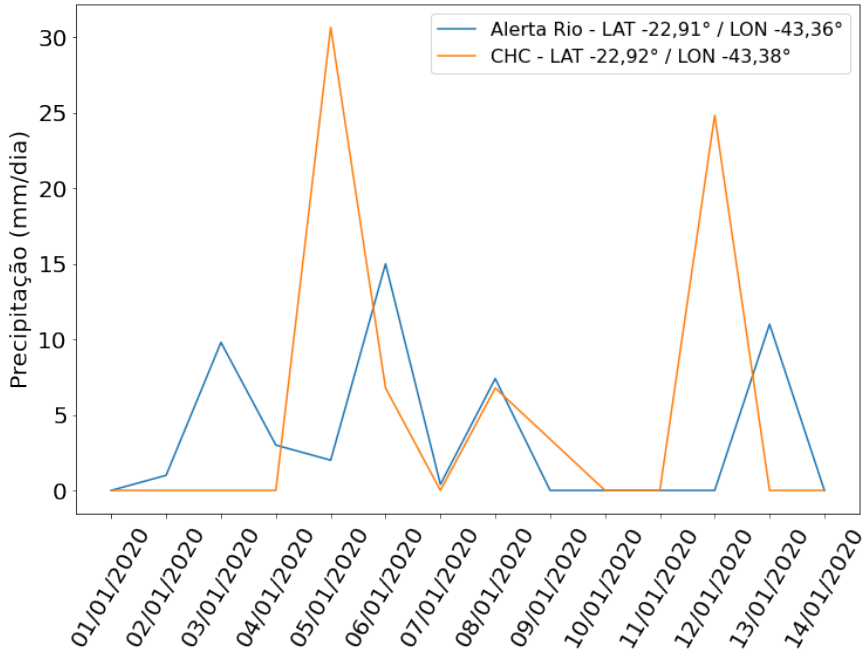


Figura I.2: Comparação entre os valores registrados por uma estação telemétrica da cidade do Rio de Janeiro e os valores interpolados do conjunto de dados do CHC. O CHC tende a superestimar os valores de precipitação.

2021; Tekin et al., 2021; Castro et al., 2021; Yang et al., 2022; Qiu et al., 2022; Ma et al., 2023]. Um ponto comum a ser ressaltado nessas abordagens é que o processo de interpolação não faz parte do treinamento do modelo de previsão. A interpolação é, portanto, uma etapa de pré-processamento dos dados.

I.3 Objetivos

Este trabalho se enquadra como subprojeto do projeto *RioNowcast* e tem por objetivo experimentar e avaliar o aprendizado da interpolação de dados de precipitação por meio de redes neurais para grafos. O modelo é treinado empregando dados de três tipos de instrumentos meteorológicos: radar meteorológico, estação meteorológica e pluviômetro. A hipótese é que a rede seja capaz de treinar um modelo de interpolação de precipitação por meio do aprendizado dos padrões da física da atmosfera, empregando séries temporais multivariadas de múltiplos instrumentos meteorológicos. Por resultado, espera-se que o modelo seja capaz de inferir valores de precipitação mais precisos para áreas não devidamente cobertas pelas estações telemétricas, dado este que tem por característica ser significativamente desbalanceado, *i.e.*, muitos dias sem chuva.

Como objetivos secundários, pretende-se: *i)* realizar testes de sensibilidade sobre os dados do radar meteorológico, *i.e.*, o quanto essa variável contribui para a acurácia do modelo; *ii)* analisar a influência que os dados da vizinhança têm sobre a interpolação de um determinado ponto no espaço. As contribuições esperadas deste trabalho são:

- Desenvolvimento de um novo método de interpolação de precipitação baseado em redes neurais para grafos, empregando múltiplas fontes de dados multivariados.
- Definição de um processo de *otimização* e *curadoria* para tratamento de dados de diferentes tipos de instrumentos meteorológicos.
- Implantação e disponibilização de um repositório de dados meteorológicos para o projeto *RioNowcast*.

I.4 Metodologia

A área territorial de estudo deste trabalho abrange a cidade do Rio de Janeiro e está delimitada conforme a Figura I.1. A primeira etapa deste trabalho consiste no processo de *Extract Transform Load* (ETL) dos dados necessários para o experimento. Para tal, é definido um processo de *otimização* e *curadoria* dos dados meteorológicos que é detalhado no Capítulo IV. São empregadas as seguintes fontes de dados, todas com seus instrumentos de observação instalados dentro da área de estudo:

- 33 estações telemétricas do sistema Alerta Rio.
- 4 estações telemétricas do Instituto Nacional de Meteorologia (INMET).
- 1 radar meteorológico do Instituto Estadual do Ambiente (INEA).

A arquitetura de rede neural para grafos é modelada de forma que os vértices denotem as coordenadas geográficas[3] e as arestas denotem uma relação entre duas coordenadas. Cada vértice é associado a um conjunto de variáveis meteorológicas para a respectiva coordenada. No contexto da física da atmosfera, é esperado que as condições climáticas de regiões próximas sejam similares e que ao longo do tempo apresentem relação de dependência. Por conta disso, as arestas são definidas por um limiar de distância entre as coordenadas. Cabe ressaltar que, neste trabalho, os vértices não necessariamente representam uma coordenada em que existe uma estação telemétrica. Tal modelagem busca atender o requisito deste trabalho que é interpolar valores de precipitação para áreas não cobertas pelas estações existentes.

Em nosso método, definimos previamente uma estação telemétrica que servirá como referência para avaliar o modelo de interpolação. Chamamos essa estação de *Estação de Interesse* (EI). A partir dessa definição, removemos do conjunto de dados todas as observações meteorológicas da EI, simulando uma coordenada sem informação e que deseja-se interpolar. O modelo é então treinado para interpolar a precipitação, empregando no treinamento os dados das *Estações Vizinhas* (EV)

[3]Neste trabalho, e comumente usado no contexto de geolocalização, os termos *coordenadas* e *ponto* são intercambiáveis no sentido que um ponto em um mapa é denotado por uma tupla de coordenadas (latitude, longitude).

e do radar meteorológico. Ao final, comparamos o valor de precipitação registrado pela EI com o valor interpolado pelo modelo para a mesma coordenada, obtendo o percentual de acurácia do modelo.

I.5 Organização

Esta dissertação está organizada como segue. O Capítulo II apresenta a fundamentação teórica e o Capítulo III descreve os trabalhos relacionados. O Capítulo IV apresenta uma visão geral do trabalho, formalizando o problema, detalhando as fontes de dados meteorológicos e a arquitetura de rede neural para grafos empregada. O Capítulo V explana sobre a análise exploratória dos dados meteorológicos empregados, o processo de construção do grafo, o pré-processamento dos dados, a validação cruzada do experimento e o treinamento do modelo. O Capítulo VI descreve os resultados e a análise sobre os mesmos. Por fim, o Capítulo VII apresenta as conclusões e os trabalhos futuros.

Capítulo II Referencial Teórico

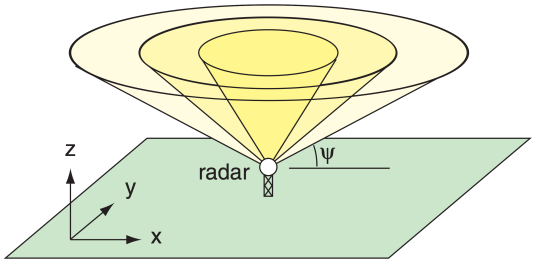
Este capítulo inicia com as Seções II.1 e II.2 que abordam as características e o funcionamento dos instrumentos meteorológicos *radar* e *estação meteorológica*. A Seção II.3 discute o processo de interpolação de dados geolocalizados e a inferência de valores pelo método de interpolação *Inverse Distance Weighting*. A Seção II.4 trata de grafos, explicando seus conceitos fundamentais. A Seção II.5 descreve as componentes e o funcionamento das *Artificial Neural Network* (ANN) e a Seção II.6 detalha a *Graph Convolutional Network* que é um tipo de ANN para inferência em estruturas de dados em forma de grafos. A Seção II.7 apresenta o conceito de séries temporais e como esse tipo de dado pode ser estruturado para treinar modelos de previsão. A Seção II.8 explica o funcionamento geral de uma *Recurrent Neural Network* (RNN), que é um tipo de ANN para aprendizado em séries temporais e detalha a *Gated Recurrent Unit*, um componente moderno empregado em RNNs. A Seção II.9 detalha o *mecanismo de atenção*, um componente moderno que pode ser empregado em diversas arquiteturas de ANN para melhorar o desempenho. Por fim, a Seção II.10 apresenta as *Spatiotemporal Graph Convolutional Network*, que é um tipo de ANN para inferência em grafos associados à séries temporais de dados.

II.1 Radar Meteorológico

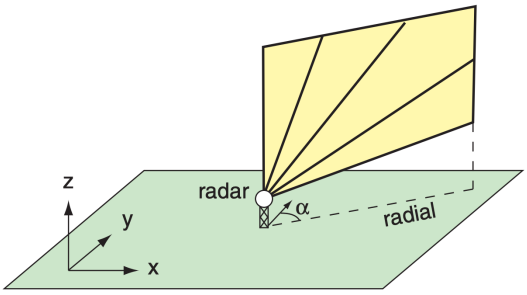
O radar meteorológico moderno é um instrumento basicamente composto de uma *antena* com *transmissor* e *receptor* e um *processador de sinal*. O transmissor irradia energia eletromagnética para a atmosfera que é refletida pelos diversos alvos. O receptor captura o sinal refletido que é processado em informação útil pelo processador de sinal. A partir do sinal refletido, é possível detectar a presença, direção, elevação, distância, intensidade e velocidade de alvos na atmosfera. O processador de sinal aplica filtros para desconsiderar alvos não desejados (*e.g.* terreno, insetos, pássaros). Após o filtro, o processador de sinal registra os alvos de interesse que são chamados de *hidrometeoros* (*e.g.* água, gelo, neve).

O radar meteorológico possui parâmetros de operação que podem ser configurados. Um subconjunto desses parâmetros define a *seção de escaneamento do radar*. Uma seção de escaneamento ocorre num intervalo de tempo que varia entre 4 e 10 minutos. Cada seção de escaneamento varre a atmosfera em A azimutes em 360 graus e Ψ ângulos de elevação. Cada azimute α é operado em

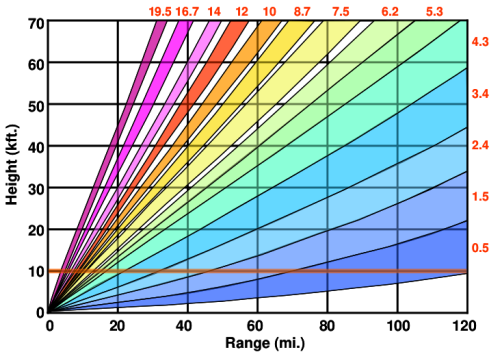
diferentes ângulos de elevação, possibilitando observar tanto distâncias quanto altitudes maiores da atmosfera. Quanto menor o ângulo de elevação da antena, maior a distância observada, porém em altitudes menores. Já em ângulos maiores de elevação, observam-se altitudes maiores, porém em distância reduzida. A Figura II.1 ilustra a forma como esse escaneamento é realizado, bem como a relação entre distâncias e altitudes observadas para cada ângulo de elevação.



(a) seção de escaneamento



(b) escaneamento em azimute fixo



(c) relação elevação/distância,altitude

Figura II.1: Escaneamento da atmosfera por radar meteorológico. (a) Cada seção de escaneamento do radar varre a atmosfera em A azimutes em 360 graus e Ψ ângulos de elevação. (b) Cada azimute α é operado em diferentes ângulos de elevação, possibilitando observar tanto distâncias quanto altitudes maiores da atmosfera. (c) Quanto menor o ângulo de elevação da antena, maior a distância observada, porém em altitudes menores. Já em ângulos maiores de elevação, observa-se altitudes maiores, porém em distância reduzida. Imagens (a),(b): https://bit.ly/libretexts-weather-radars. Imagem (c): https://bit.ly/ucla-weather-radar-fundamentals.

Um *feixe de energia* do radar é a transmissão de uma onda eletromagnética em um determinado azimute e ângulo (α_i, ψ_j). A Figura II.2 ilustra a emissão de um feixe de energia do radar. O feixe é segmentado pelo processador de sinais em volumes chamados *bins*. O volume iluminado é a menor informação identificável no feixe do radar. Devido ao feixe ter formato de cone, o volume aumenta conforme aumenta a distância do radar.

O processador de sinais se encarrega de derivar variáveis meteorológicas (*e.g. refletividade*, *velocidade radial*) a partir das informações dos volumes iluminados. A variável meteorológica *refletividade* é determinada pelo somatório dos diâmetros dos hidrometeoros dentro de um volume iluminado. Dessa forma, é possível estimar a quantidade total de hidrometeoros em um determinado

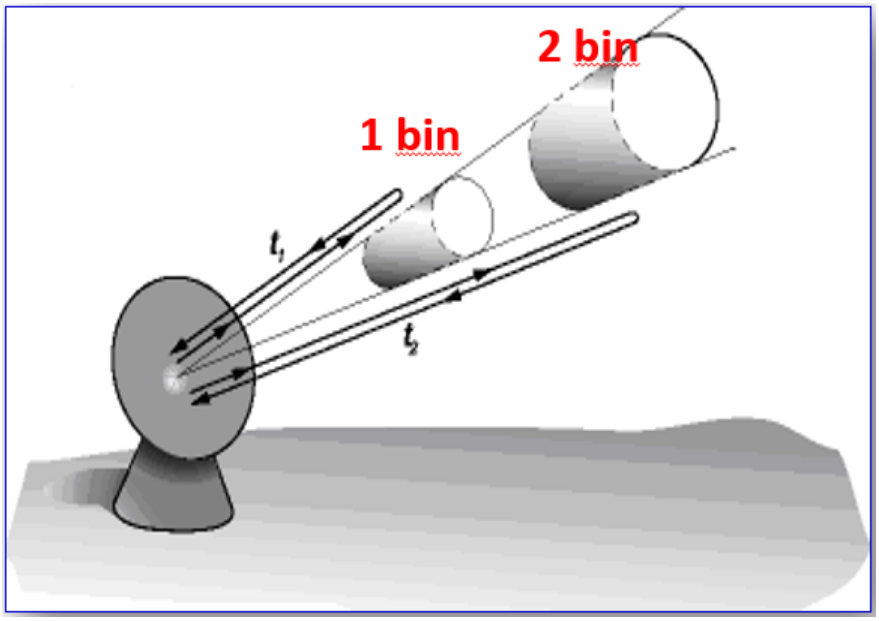


Figura II.2: Feixe de energia do radar meteorológico. O radar transmite um onda eletromagnética que é segmentada em volumes ou *bins*. O volume aumenta conforme aumenta a distância do radar. O processador de sinais do radar calcula as propriedades médias dos hidrometeoros em cada volume. Imagem: Curso Radar Meteorológico do CPTEC/INPE.

volume.

A forma de operação do radar, também chamada de *estratégia de escaneamento*, causa um efeito chamado de *cone do silêncio*. O cone do silêncio é a porção em torno e acima do radar que não é observada. A Figura II.3 apresenta uma estratégia de escaneamento com quatorze ângulos de elevação. Considerando todos esses ângulos, é possível identificar na figura a área em torno e acima do radar que não é coberta pelos feixes de energia.

II.2 Estações Meteorológicas

O termo *Estação Meteorológica* é normalmente empregado para denotar um instrumento telemétrico instalado em superfície. *Telemetria* tem origem Grega (*tele* = remoto e *metron* = medida) e refere-se à uma tecnologia que permite a medição e comunicação de informações. Na literatura encontramos frequentemente termos sinônimos para denotar estações meteorológicas, como por exemplo *estações telemétricas*, *estações de superfície* ou *estações meteorológicas*. Neste trabalho definimos os seguintes termos e definições:

- *Estação Meteorológica*: uma estação que mede múltiplas variáveis meteorológicas.
- *Estação Pluviométrica* ou *Pluviômetro*: uma estação que mede somente a variável meteorológica *precipitação*.
- *Estação Telemétrica*: uma estação meteorológica ou um pluviômetro.

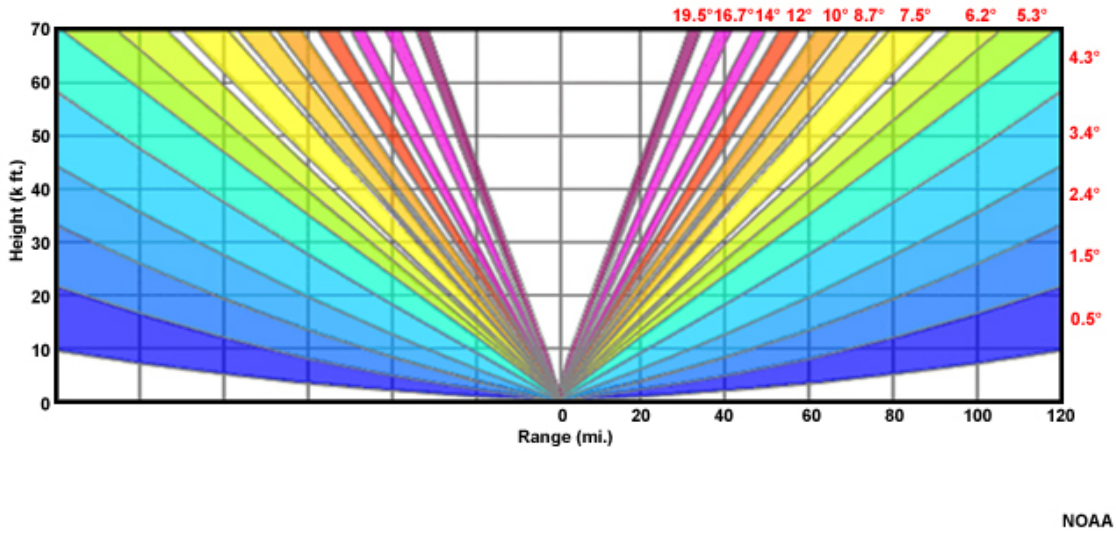


Figura II.3: Cone do silêncio. Uma porção da área acima e em torno do radar não é observada. O tamanho da porção varia conforme o ângulo de elevação do radar. Imagem: `https://bit.ly/ucla-weather-radar-fundamentals`.

Estações telemétricas são dotadas de sensores responsáveis em observar variáveis meteorológicas. Cada tipo de sensor observa uma variável, logo é possível existir um ou mais sensores em uma mesma estação telemétrica. Não há uma regra sobre a região espacial que uma estação telemétrica representa em termos de valores observados. Isso pode variar dependendo da topologia da localização, o que requer uma análise específica para cada caso. Dentre as variáveis meteorológicas comumente observadas por estações telemétricas, podemos citar: *precipitação*, *temperatura*, *umidade relativa do ar*, *direção do vento*, *velocidade do vento*, dentre outras.

II.3 Interpolação de Dados Geolocalizados

No contexto de dados de estações telemétricas, o processo de interpolação pode ser entendido como a inferência de valores para as localizações da área territorial não cobertas por pelo menos uma estação. Em uma determinada área territorial é esperado que as condições climáticas em subáreas próximas sejam similares e, portanto, é admissível obter uma estimativa em determinada coordenada geográfica (latitude x longitude) a partir dos dados meteorológicos observados em coordenadas próximas. Dessa forma, é possível transformar uma grade irregular de dados observados em uma grade regular. Em geral os métodos de interpolação processam os valores de acordo com a Equação II.1

$$x_i = \sum_{j=1}^{N} \lambda_j x_j, \tag{II.1}$$

em que x_i é o valor interpolado para uma determinada coordenada da grade de dados, N é o número de estações telemétricas, x_j é o valor mensurado pela estação telemétrica j e λ_j é o peso associado à estação telemétrica j. A diferença entre os métodos reside na forma como λ_j é definido [Lyra et al., 2018].

Uma variação da Equação II.1 é chamada *Inverse Distance Weighting* (IDW). No método IDW λ_j é denotado pela Equação

$$\lambda_j = (1/d_{ij}^p)/\sum_{j=1}^{N}(1/d_{ij}^p), \tag{II.2}$$

em que d_{ij} é a distância entre as estações i e j e p é a potência usada na interpolação. O valor mais comum para p é 2 o qual torna IDW um estimador baseado na distância quadrada inversa. Da forma como λ_j é definido e considerando que x_i é calculado com base nos valores de m estações sendo $1 \leq m \leq N$, então

$$\sum_{j=1}^{m} \lambda_j = 1. \tag{II.3}$$

II.4 Grafos

Grafo é uma estrutura matemática capaz de representar objetos e as relações entre estes. Formalmente, um grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ é definido por um conjunto de vértices $v \in \mathcal{V}$ (objetos) e um conjunto de arestas $e \in \mathcal{E}$ (relações). Vértices e/ou arestas podem possuir atributos que denotem determinados aspectos do domínio que se deseja modelar. Os vértices de um grafo podem estar associados ao mesmo conjunto de atributos. Neste caso, dizemos que o grafo é **homogêneo**. No entanto, existem grafos em que os vértices representam entidades distintas do domínio e por isso podem estar associados a conjuntos distintos de atributos. Quando este é o caso, dizemos que o grafo é **heterogêneo**.

Considere como exemplo um grafo que representa uma área territorial. O grafo pode ter seus vértices atribuídos de um índice único que denote uma determinada coordenada (latitude, longitude), bem como um vetor de valores observados para a coordenada (*e.g.* precipitação, temperatura, umidade do ar). Suas arestas podem ter um valor atribuído que represente a distância entre um par de coordenas. Esse recurso é largamente utilizado em grafos e por meio dele vários problemas do mundo real podem ser computacionalmente solucionados.

Em um grafo, dois vértices $v_i, v_j \in \mathcal{V}$ são ditos **adjacentes** (ou seja, há relação entre os objetos) quando existe uma aresta $e_{ij} \in \mathcal{E}$. Vértices adjacentes são ditos **vizinhos**. O número de arestas incidentes em um vértice é chamado **grau**. É possível existir uma aresta que relacione um vértice a ele próprio. Nesse caso, denotamos que, em relação ao vértice v_i, existe uma aresta $e_{ii} \in \mathcal{E}$.

Chamamos essa aresta de **auto-laço**. Arestas podem denotar a direção de uma relação. Nesse caso, dizemos que o grafo é **direcionado**. Em um grafo direcionado, podemos ter $e_{ij} \neq e_{ji}$.

Uma das formas de se representar um grafo computacionalmente é por meio de uma **matriz de adjacência**. A matriz de adjacência **A** de um grafo com n vértices possui dimensões $n \times n$. Uma entrada $a_{i,j}$ (linha i, coluna j) é igual a 1 se existe uma aresta entre os vértices v_i e v_j, 0 caso contrário. A diagonal principal da matriz é toda entrada $a_{i,j}$, com $i = j$. Em grafos sem auto-laços a diagonal principal é sempre igual a 0. Um grafo não direcionado é representado por uma matriz de adjacência que é **simétrica**: $a_{i,j} = a_{j,i}\ \forall i, j$. A Figura II.4 mostra um exemplo de grafo não direcionado e sua respectiva matriz de adjacência.

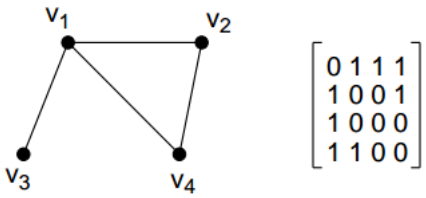


Figura II.4: Matriz de Adjacência. Lehman and Leighton [2004]

II.5 Artificial Neural Network

Artificial Neural Network (ANN) (em português, *Rede Neural Artificial*) é uma arquitetura computacional baseada no cérebro biológico que tem por objetivo processar informação. O surgimento das ANNs pode ser atribuído à complexidade de implementação de sistemas para resolução de problemas complexos baseados em regras heurísticas. Como forma de transpor tal dificuldade, as ANNs são implementadas de forma a receber como entrada dados de exemplo e por meio de sucessivas observações aprender determinados padrões sobre eles. Tal etapa é comumente chamada de *treinamento* e baseia-se no aprendizado humano que evolui por meio das experiências. Os padrões aprendidos geram um modelo capaz de inferir sobre dados não utilizados durante o treinamento, mas que possuam padrões similares. A seguir, são descritos os componentes e o funcionamento das ANNs que foram exploradas neste trabalho, que têm como fonte o tutorial de aprendizado profundo da Universidade de Stanford [Stanford, 2019].

II.5.1 Perceptron

A unidade básica de processamento de uma rede neural artificial é chamada de *perceptron*. O *perceptron* foi proposto por Rosenblatt [1958] como aprimoramento do trabalho anterior de McCulloch and Pitts [1943] e sua estrutura foi inspirada no neurônio biológico. A Figura II.5 mostra um exemplo de *perceptron*.

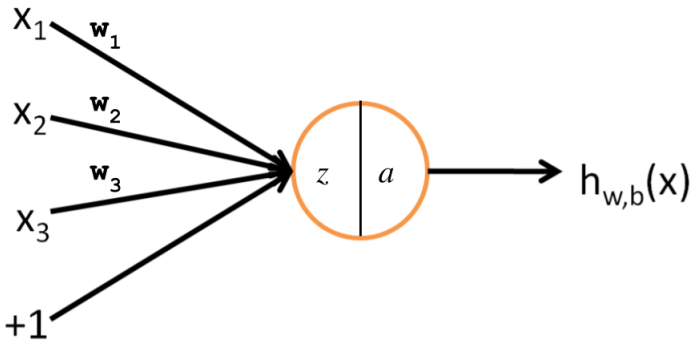


Figura II.5: Perceptron. Stanford [2019]

Considerando o perceptron da Figura II.5, os valores x_1, x_2 e x_3 são passados como entrada. Cada um desses valores é multiplicado por seus respectivos pesos w_1, w_2 e w_3 e a soma destes produtos com o termo de interceptação b $(+1)$ gera o valor agregado z, que sofre a transformação por meio da função $f(\cdot)$ gerando o valor a. Esse processamento é resumido pela Equação II.4

$$h_{\mathbf{w},b}(\mathbf{x}) = f(\mathbf{w}^T\mathbf{x}) = f(\sum_{i=1}^{3} w_i x_i + b), \tag{II.4}$$

onde $f : \mathbb{R} \to \mathbb{R}$ é chamada de *função de ativação*. As funções de ativação $f(\cdot)$ comumente utilizadas são a *sigmoid* (Equação II.5) que tem como resultado valores entre $[0, 1]$, a *hyperbolic tangent* (Equação II.6) que tem como resultado valores entre $[-1, 1]$ e a *Rectified Linear Unit* (ReLU) (Equação II.7) que retorna o maior valor entre $(0, z)$.

$$f(z) = \frac{1}{1 + \exp(-z)}, \tag{II.5}$$

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \tag{II.6}$$

$$f(z) = \max(0, z). \tag{II.7}$$

Denotamos $h_{\mathbf{w},b}(\mathbf{x})$ a hipótese que o *perceptron* calcula tendo como entrada $\mathbf{x} = \{x_1, x_2, x_3\}$, utilizando-se dos parâmetros $\mathbf{w}$ e b que devem ser aprendidos. Denotamos $\mathbf{x}^{(i)}$ como o i-ésimo exemplo de um conjunto de dados e $y^{(i)}$ o seu respectivo valor alvo, o qual chamamos de *rótulo*. Em uma tarefa de classificação podemos definir $y^{(i)} = 1$ se $\mathbf{x}^{(i)}$ pertence a uma determinada classe arbitrária, 0 caso contrário. Como unidade única de processamento, o *perceptron* somente é capaz de aprender funções lineares. Para o aprendizado de funções não lineares, as unidades *perceptrons* precisam ser agrupadas em camadas de processamento formando uma arquitetura chamada *Multi-layer Perceptron* (MLP).

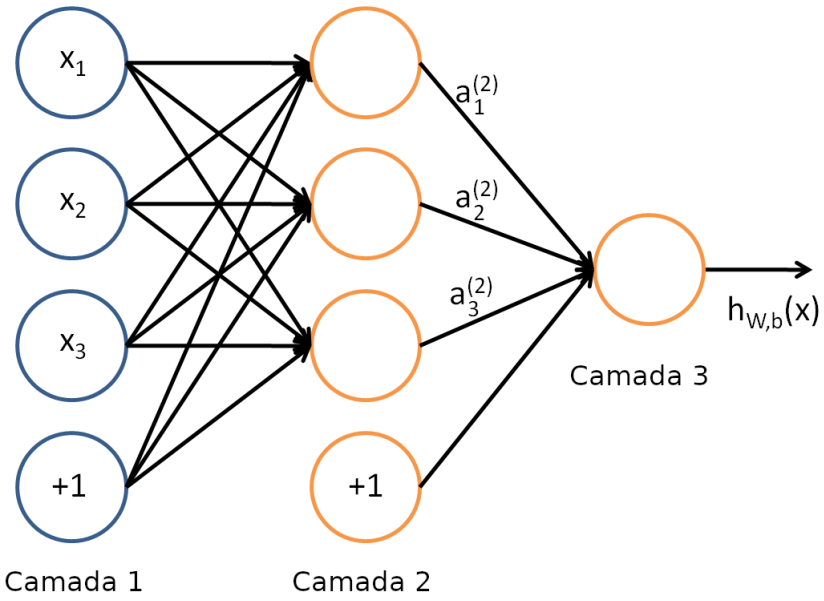


Figura II.6: Multilayer Perceptron. Stanford [2019]

II.5.2 Multilayer Perceptron

Para o aprendizado de funções não lineares, os *perceptrons* são agrupados em camadas. Tal arquitetura é organizada de forma que a saída de um neurônio seja a entrada de outro neurônio da camada seguinte. A Figura II.6 mostra um exemplo de MLP.

Neste exemplo, denotamos $w_{ij}^{(l)}$ como o peso associado à conexão entre a unidade j na camada l e a unidade i na camada $l+1$, com $l = \{1, 2, ..., L\}$. Denotamos o termo de interceptação associado à unidade i na camada $l+1$ por $b^{(l)}$ o qual, no contexto das MLPs, é chamado *bias*. Denotamos $a_i^{(l)}$ como a ativação da unidade i na camada l. Para $l = 1$, usamos $a_i^{(1)} = x_i$ para denotar a i-ésima entrada. A computação realizada pela MLP é então representada pelas Equações II.8, II.9, II.10 e II.11.

$$a_1^{(2)} = f(w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2 + w_{13}^{(1)}x_3 + b^{(1)}) \tag{II.8}$$

$$a_2^{(2)} = f(w_{21}^{(1)}x_1 + w_{22}^{(1)}x_2 + w_{23}^{(1)}x_3 + b^{(1)}) \tag{II.9}$$

$$a_3^{(2)} = f(w_{31}^{(1)}x_1 + w_{32}^{(1)}x_2 + w_{33}^{(1)}x_3 + b^{(1)}) \tag{II.10}$$

$$h_{W,b}(x) = a_1^{(3)} = f(w_{11}^{(2)}a_1^{(2)} + w_{12}^{(2)}a_2^{(2)} + w_{13}^{(2)}a_3^{(2)} + b^{(2)}) \tag{II.11}$$

Denotamos $z_i^{(l)}$ como a soma ponderada da unidade i na camada l, incluindo o *bias*. Por exemplo, $z_i^{(2)} = \sum_{j=1}^{N} w_{ij}^{(1)}x_j + b^{(1)}$, de modo que $a_i^{(l)} = f(z_i^{(l)})$. Considerando a função de ativação $f(\cdot)$ aplicada a vetores na forma elemento a elemento (*i.e.*, $f([z_1^{(l+1)}, z_2^{(l+1)}, z_3^{(l+1)}]) = [f(z_1^{(l+1)}), f(z_2^{(l+1)}), f(z_3^{(l+1)})]$), chamamos de *propagação adiante* a computação realizada pela MLP

que pode ser então simplificada pelas Equações II.12 e II.13.

$$\mathbf{z}^{(l+1)} = \mathbf{w}^{(l)}\mathbf{a}^{(l)} + b^{(l)} \tag{II.12}$$

$$\mathbf{a}^{(l+1)} = f(\mathbf{z}^{(l+1)}) \tag{II.13}$$

Essa classe de rede é conhecida como *Feed Forward Neural Network* (FFNN). Apesar do exemplo simples, FFNNs podem variar em número de camadas, número de unidades por camada e até mesmo número de unidades de saída. Em uma arquitetura onde queremos que a rede calcule a probabilidade de um exemplo $\mathbf{x}^{(i)}$ pertencer a um número N de classes distintas, podemos ter $\mathbf{y}^{(i)} \in \mathbb{R}^N$.

O treinamento de uma FFNN consiste então em aprender os valores de $(\mathbf{W}, \mathbf{b})$, de modo que uma entrada $\mathbf{x}^{(i)}$ seja mapeada pela rede para o rótulo $y^{(i)}$. Vale ressaltar que para cada camada (à exceção da camada L), existe um par $(\mathbf{w}^{(l)}, b^{(l)})$. O gradiente descendente é utilizado nessa etapa do treinamento chamada *retropropagação*, a qual atualiza os valores de $(\mathbf{W}, \mathbf{b})$ conforme $h_{\mathbf{W},\mathbf{b}}(\mathbf{x}^{(i)})$ se aproxima ou se afasta de $y^{(i)}$. A distância entre esses dois valores é chamada de *custo* e é denotada pela Equação II.14.

$$J(\mathbf{W}, \mathbf{b}; \mathbf{x}, y) = \frac{1}{2} \left\| h_{\mathbf{W},\mathbf{b}}(\mathbf{x}) - y \right\|^2 \tag{II.14}$$

Considerando um conjunto de dados com M exemplos, o custo total da rede pode ser calculado pelas Equações II.15 e II.16.

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{M} \sum_{i=1}^{M} J(\mathbf{W}, \mathbf{b}; \mathbf{x}^{(i)}, y^{(i)}) \tag{II.15}$$

$$= \frac{1}{M} \sum_{i=1}^{M} \left(\frac{1}{2} \left\| h_{\mathbf{W},\mathbf{b}}(\mathbf{x}^{(i)}) - y^{(i)} \right\|^2 \right) \tag{II.16}$$

O objetivo do treinamento é então minimizar $J(\mathbf{W}, \mathbf{b})$. O processo de treinamento inicia pela atribuição de valores aleatórios próximos de zero para cada parâmetro $w_{ij}^{(l)}$ e $b^{(l)}$. Após executar a propagação adiante com os M exemplos do conjunto de dados, $J(\mathbf{W}, \mathbf{b})$ é calculado. A partir deste ponto, inicia-se a retropropagação que nada mais é do que a etapa de atualização dos parâmetros. A atualização é dada pelas Equações II.17 e II.18

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \alpha \frac{\partial}{\partial w_{ij}^{(l)}} J(\mathbf{W}, \mathbf{b}) \tag{II.17}$$

$$b^{(l)} = b^{(l)} - \alpha \frac{\partial}{\partial b^{(l)}} J(\mathbf{W}, \mathbf{b}), \tag{II.18}$$

onde α é chamado de *taxa de aprendizado* e é considerado um hiperparâmetro que deve ser ajustado por meio de alguma heurística e a derivada da função de custo $J(\mathbf{W}, \mathbf{b})$ é calculada pelas Equações II.19 e II.20.

$$\frac{\partial}{\partial w_{ij}^{(l)}} J(\mathbf{W}, \mathbf{b}) = \frac{1}{M} \sum_{i=1}^{M} \frac{\partial}{\partial w_{ij}^{(l)}} J(\mathbf{W}, \mathbf{b}; \mathbf{x}^{(i)}, y^{(i)}) \tag{II.19}$$

$$\frac{\partial}{\partial b^{(l)}} J(\mathbf{W}, \mathbf{b}) = \frac{1}{M} \sum_{i=1}^{M} \frac{\partial}{\partial b^{(l)}} J(\mathbf{W}, \mathbf{b}; \mathbf{x}^{(i)}, y^{(i)}) \tag{II.20}$$

Excluindo a etapa inicial de atribuição aleatória dos parâmetros, a rede repete a propagação adiante e a retropropagação até que haja a convergência. Entende-se por convergência o valor de $J(\mathbf{W}, \mathbf{b})$ chegar a um valor mínimo tal que novas iterações modifiquem irrisoriamente o valor de $J(\mathbf{W}, \mathbf{b})$.

II.6 Graph Convolutional Network

Graph Convolutional Network (GCN) é uma arquitetura de ANN que possibilita realizar tarefas de inferência em grafos (*e.g.* predição, classificação). A ideia principal por trás das GCNs consiste em aplicar processos de convolução para gerar uma *representação* (também chamado de *embedding*) para cada vértice do grafo. Em geral, as representações são passadas como entrada para camadas MLP que processam a tarefa de inferência. As formulações a seguir consideram o cenário de grafos homogêneos. O aprendizado em grafos heterogêneos não é escopo deste trabalho.

Considere que cada vértice do grafo esteja associado a um vetor de características $\mathbf{x} \in \mathbb{R}^F$. O processo de convolução em uma GCN consiste na combinação do vetor de características de um vértice com a agregação das características dos vértices vizinhos. Em uma GCN com L camadas convolucionais, sendo $l = \{1, 2, \cdots, L\}$, a convolução de um vértice pode ser representada pelas Equações II.21 e II.22

$$\mathbf{h}_{N(v)}^{l} = f(\{\mathbf{h}_{u}^{l-1}, \forall u \in N(v)\}) \tag{II.21}$$

$$\mathbf{h}_v^l = \sigma(\mathbf{w}^l \cdot [\mathbf{h}_v^{l-1}, \mathbf{h}_{N(v)}^l]), \tag{II.22}$$

em que $N(v)$ é o conjunto de vértices vizinhos de v, $f(\cdot)$ é uma função de agregação arbitrária, $[\mathbf{h}_v^{l-1}, \mathbf{h}_{N(v)}^l]$ é a combinação dos vetores, σ é uma função não linear e $\mathbf{w}$ é a matriz de pesos que devem ser aprendidos [Wu et al., 2019; Hamilton et al., 2017].

Kipf and Welling [2016] desenvolveram uma regra de propagação que codifica a estrutura do grafo em um modelo de rede neural $f(\mathbf{X}, \mathbf{A})$, sendo $\mathbf{X}$ a matriz de *features* associadas aos vértices e $\mathbf{A}$ a matriz de adjacência do grafo. Condicionar $\mathbf{A}$ a $f(.)$, permite que o modelo realize o treino tanto de forma supervisionada quanto semissupervisionada. Além disso, a regra proposta habilita o aprendizado de representações para todos os vértices, *i.e.*, vértices com e sem rótulos associados. Por fim, a operação é modelada como um produto de matrizes, tornando-a computacionalmente eficiente.

Conforme proposto por Kipf and Welling [2016], a GCN processa $f(\mathbf{X}, \mathbf{A})$ sobre L camadas de convolução denotada pela Equação

$$\mathbf{H}^{(l+1)} = \sigma(\hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{II.23}$$

em que $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ é a matriz de adjacência com auto-laços. $\mathbf{I}_N$ é a matriz identidade, $\hat{\mathbf{D}} = \hat{\mathbf{D}}_{ii} = \Sigma_j \hat{\mathbf{A}}_{ij}$ é a matriz diagonal de graus e $\mathbf{W}^{(l)}$ é a matriz de pesos da camada l. $\sigma(\cdot)$ denota uma função de ativação. $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times D}$ é a matriz de ativações da l-ésima camada. $\mathbf{H}^{(0)} = \mathbf{X}$, sendo $\mathbf{X} \in \mathbb{R}^{N \times F}$ a matriz de *features* associadas aos vértices.

O *campo receptivo* de um vértice v_i é o conjunto de vértices que contribuem para a determinação de sua representação final. O campo receptivo de v_i cresce a medida que são empilhadas mais camadas convolucionais. Em outras palavras, conforme empilhamos camadas convolucionais, vértices vizinhos cada vez mais distantes passam a compor o campo receptivo de v_i. Ao contrário de outros tipos de ANNs, empilhar camadas convolucionais em demasia em uma GCN pode levar a convergência das representações dos vértices, *i.e.*, todos os vértices do grafo convergem para o mesmo valor. Isso se torna um problema, uma vez que queremos que as representações diferenciem os vértices. Uma das formas de evitar tal problema é analisar o campo receptivo necessário para resolver a tarefa de aprendizado no domínio dos dados.

II.7 Séries Temporais

No contexto de dados, *séries temporais* são sequências de valores registrados ao longo do tempo. Séries temporais são categorizadas em dois tipos: *univariada* e *multivariada*. Uma série univariada $\mathbf{s}$ pode ser denotada pela expressão $\mathbf{s} = \{s_1, s_2, \cdots, s_T\} \in \mathbb{R}^T$, em que $s_t \in \mathbb{R}$ é um valor escalar,

sendo $t = \{1, 2, \cdots, T\}$. Já em uma série multivariada $\mathbf{s}_t \in \mathbb{R}^N$, *i.e.*, $\mathbf{s}_t$ é um vetor contendo N variáveis [Jin et al., 2024]. Por exemplo, uma série multivariada poderia conter as observações de uma estação meteorológica ao longo do tempo. Para cada instante de tempo, a estação poderia registrar um vetor contendo os valores observados para as variáveis *precipitação*, *temperatura* e *umidade relativa do ar*.

Dados em séries temporais podem ser empregados para treinar um modelo de previsão usando uma ANN. Em geral, treinamos o modelo para inferir um ou mais valores futuros a partir de uma sequência de valores observados até o momento presente. Por exemplo, a partir da série $\{s_{T-2}, s_{T-1}, s_T\}$, o modelo $f(\cdot)$ pode ser treinado para prever os valores futuros $\{\hat{s}_{T+1}, \hat{s}_{T+2}, \cdots\}$. Quando o modelo é treinado para prever o valor em um único passo de tempo futuro (escalar ou vetorizado), dizemos que a tarefa é do tipo *single-step forecasting*. Quando a tarefa envolve prever vários passos de tempo futuros, dizemos que é do tipo *multi-step forecasting*.

Para treinar um modelo de previsão para séries temporais, é preciso garantir que os dados da série temporal estejam devidamente estruturados para servirem como exemplos de treinamento. Uma técnica comumente empregada é chamada de *janelas deslizantes*. O Algoritmo 1 apresenta o processo de construção dos exemplos de treinamento empregando esta técnica. Para fins de simplificação, assumimos a tarefa *single-step forecasting* em uma série univariada, apesar da técnica ser perfeitamente adaptável para *multi-step forecasting* e multivariada. Além disso, consideramos que a série temporal é contígua no tempo. Esta questão será melhor detalhada mais abaixo no texto.

A técnica é processada como se segue. Dado uma série temporal $\mathbf{s}$, define-se o tamanho das janelas w_{in} e w_{out} que representam, respectivamente, o tamanho da série temporal que é dada como entrada para o modelo e o tamanho da série temporal que o modelo deve inferir. A partir dos valores w_{in} e w_{out}, o algoritmo itera sobre os indexadores de $\mathbf{s}$ extraindo tuplas de exemplos de treinamento, cada tupla contendo as *features* e os rótulos para um treinamento supervisionado. Considerando $\mathbf{s} = \{10, 20, 30, 40, 50\}$, $w_{in} = 3$ e $w_{out} = 1$, o Algoritmo 1 retorna como resultado as tuplas estruturadas na Tabela II.1 para facilitar a visualização.

Algorithm 1 Processo de Janelamento para Séries Temporais

Require: Série temporal $\mathbf{s} = \{s_1, s_2, \ldots, s_T\}$, tamanho da janela de input w_{in}, tamanho da janela de output w_{out}

```
T ← length(s)
train_examples ← []
for i = 1 to T − w_in − w_out + 1 do
    input_window ← s[i : i + w_in − 1]
    output_window ← s[i + w_in : i + w_in + w_out − 1]
    train_examples.append((input_window, output_window))
end for
return train_examples
```

Tabela II.1: Exemplos criados pelo processo de janelamento para a série $\mathbf{s} = \{10, 20, 30, 40, 50\}$, $w_{in} = 3$ e $w_{out} = 1$.

***Features* de Entrada**	**Rótulo**
[10, 20, 30]	[40]
[20, 30, 40]	[50]

Uma questão importante na aplicação da técnica *janelas deslizantes* reside na possibilidade da série temporal não ser contígua no tempo. Considere uma estação pluviométrica que registra o valor de precipitação a cada 1 hora. Considere que a estação, devido a problemas técnicos, parou de funcionar por um determinado período. Considere que a série temporal $\mathbf{s} = \{(T - 20, p_{20}), (T - 19, p_{19}), (T - 18, p_{18}), (T - 2, p_2), (T - 1, p_1), (T, p)\}$ abrange o período em que houve interrupção de operação da estação, sendo cada tupla $(T - i, p_i)$ a data/hora da observação e o valor de precipitação observado. Aplicar o Algoritmo 1 sobre $\mathbf{s}$ com parâmetros $w_{in} = 3$ e $w_{out} = 1$ resultaria como primeiro exemplo de treinamento a tupla $((p_{20}, p_{19}, p_{18}), (p_2))$. Este exemplo não é válido, já que p_2 ocorre 16 horas após p_{18}. Assim sendo, é preciso garantir que o Algoritmo 1 seja executado em blocos contíguos em séries temporais.

II.8 Gated Recurrent Unit

Recurrent Neural Network (RNN) é um tipo de ANN que pode ser empregada para treinar modelos de previsão em séries temporais. Em geral, esse tipo de rede itera sobre uma série temporal $\mathbf{s} = \{s_1, s_2, \cdots, s_T\}$ e a cada passo de tempo t gera uma representação $\mathbf{h}_t$ que é calculada pelo *codificador* em função de $(\mathbf{h}_{t-1}, s_t)$. Esse processamento é encapsulado em uma célula chamada *célula recorrente* e pode ser definido como $\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{s}_t)$, em que f é uma função de ativação [Cho, 2014]. No contexto de RNNs, o vetor $\mathbf{h}$ é frequentemente chamado de *hidden state*, uma vez que representa o estado contextual de uma série. Cada elemento do vetor $\mathbf{h}$, *i.e.*, cada $h_i \in \mathbf{h}$ é chamado de *hidden unit*.

Ao longo dos anos esse tipo de rede foi sendo aprimorada, resolvendo principalmente questões de desaparecimento e explosão de gradiente e eficiência de processamento. Um desses aprimoramentos foi proposto por Cho [2014] que define a *Gated Recurrent Unit* (GRU), uma célula recorrente mais simples e com menos parâmetros do que seus antecessores. A GRU emprega o conceito de portas denominadas *update gate* e *reset gate* que controlam, respectivamente, o quanto do contexto anterior deve ser adicionado e esquecido. O processamento em uma GRU é definido pelas Equações II.24, II.25, II.26, II.27

$$r_j = \sigma\left(\left[\mathbf{W}_r \mathbf{s}_t\right]_j + \left[\mathbf{U}_r \mathbf{h}_{(t-1)}\right]_j\right) \tag{II.24}$$

$$z_j = \sigma\left(\left[\mathbf{W}_z \mathbf{s}_t\right]_j + \left[\mathbf{U}_z \mathbf{h}_{(t-1)}\right]_j\right) \tag{II.25}$$

$$\tilde{h}_j^{(t)} = \phi\left(\left[\mathbf{W}\mathbf{s}_t\right]_j + \left[\mathbf{U}\left(\mathbf{r} \odot \mathbf{h}_{(t-1)}\right)\right]_j\right) \tag{II.26}$$

$$h_j^{(t)} = z_j h_j^{(t-1)} + (1 - z_j)\tilde{h}_j^{(t)}, \tag{II.27}$$

em que j representa o j-ésimo elemento do vetor, $\mathbf{W}$ e $\mathbf{U}$ são as matrizes de pesos que devem ser aprendidos, $\sigma(\cdot)$ e $\phi(\cdot)$ são, respectivamente, as funções *sigmoid* e *hyperbolic tangent*.

Tal formulação permite que o contexto anterior seja esquecido à medida que r_j, a *reset gate*, se aproxima de zero. Da mesma forma, z_j, a *update gate*, controla o quanto do contexto anterior será agregado à informação de entrada, *i.e.*, $\mathbf{s}_t$. Uma vez que cada j-ésimo elemento do vetor está associado a valores separados de *reset* e *update*, cada j-ésimo elemento incorpora dependências temporais em diferentes escalas, *i.e.*, de curto e longo prazo [Cho, 2014].

II.9 Attention Mechanism

O *attention mechanism* é um mecanismo que foi originalmente proposto por Bahdanau [2014] para melhorar o desempenho de modelos de tradução. Um modelo de tradução $f(\mathbf{x}; \mathbf{W})_{\Gamma \to P}$ traduz uma sentença $\mathbf{x}$ no idioma Γ para o idioma P, usando os pesos W. Desde então, o *attention mechanism* foi aprimorado em alguns tipos distintos tais como o *global attention*, *local attention*, *self-attention*, dentre outros e passou a ser aplicado em diversas outras tarefas de previsão.

A principal motivação para a criação do mecanismo foi resolver o gargalo de ter que codificar toda a sentença de entrada em um único vetor de tamanho fixo. Com o *attention mechanism*, Bahdanau [2014] permitiu que o modelo buscasse automaticamente as partes da sentença de entrada que são mais relevantes para traduzir cada palavra da sentença de saída, evitando codificar todas as partes em um tamanho fixo. A seguir, é descrito o mecanismo *soft-attention* proposto por Bahdanau [2014].

Considere uma sentença de entrada $\mathbf{x} \in \mathbb{R}^N$ já codificada em uma matriz $\mathbf{H'} \in \mathbb{R}^{N \times D}$ de *hidden states*, sendo D o tamanho do vetor $\mathbf{H}'_i$. Considere $\mathbf{s}_t$ o contexto atual do decodificador, sendo $\mathbf{s}_0$ uma inicialização aleatória ou até mesmo um vetor de zeros. O processo consiste em gerar uma *hidden state* em função de $\mathbf{s}_t$ e $\mathbf{H'}$ que será empregada pelo decodificador para predizer a próxima palavra na tradução. Para cada palavra a ser gerada pelo decodificador, o *soft-attention* se encarrega de calcular os pesos que ajustam a relevância de cada $\mathbf{h'}_i \in \mathbf{H'}$. O processo é formalizado pelas Equações II.28, II.29, II.30 e II.31

$$\mathbf{e}_i = \mathbf{s}_t \cdot \mathbf{H}'_i \tag{II.28}$$

$$\boldsymbol{\alpha}_i = \frac{\exp(\mathbf{e}_i)}{\sum_{j=1}^{L} \exp(\mathbf{e}_j)} \tag{II.29}$$

$$\mathbf{c} = \sum_{i=1}^{L} \boldsymbol{\alpha}_i \mathbf{H}'_i \tag{II.30}$$

$$\mathbf{h}_t = [\mathbf{c}, \mathbf{s}_t] \tag{II.31}$$

em que $\mathbf{e}_i$ consiste em uma pontuação para cada *hidden state* $\mathbf{h}'_i \in \mathbf{H}'$ em função de $\mathbf{s}_t$. Em seguida é aplicada a função *softmax* que transforma $\mathbf{e}$ em uma distribuição de probabilidades, *i.e.*, os pesos de atenção $\boldsymbol{\alpha}$. O vetor de contexto $\mathbf{c}$ é calculado por meio da média ponderada de $\mathbf{H}'_i$ com os pesos $\boldsymbol{\alpha}$. Por fim, a *hidden state* da iteração é formada pela combinação de $\mathbf{c}$ e $\mathbf{s}_t$ que será empregada pelo decodificador para predizer a próxima palavra.

II.10 Spatiotemporal Graph Convolutional Network

Spatiotemporal Graph Convolutional Network (STGCN) é uma arquitetura de ANN que possibilita realizar tarefas de inferência em grafos, em que os dados tem a forma de séries temporais. Neste cenário, o modelo recebe uma série temporal de grafos como entrada e gera representações para os vértices que incorporam não só as relações espacias como também temporais dos dados. Ainda neste cenário, pode ocorrer que a própria topologia do grafo varie com o tempo. Em outras palavras, vértices e arestas do grafo podem ser adicionados ou removidos ao longo do tempo.

Quando a topologia do grafo não varia com o tempo, dizemos que o grafo é *estático*. Quando ocorre variação, dizemos que o grafo é *dinâmico*. Podemos denotar uma série temporal $\mathcal{G}$ de grafos como $\{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_T\}$ em que $\mathcal{G}_t = (\mathbf{A}, \mathbf{X}_t)$ quando o grafo é estático e $\mathcal{G}_t = (\mathbf{A}_t, \mathbf{X}_t)$ quando o grafo é dinâmico, sendo $\mathbf{A}$ a matriz de adjacência e $\mathbf{X}$ a matriz de *features*. A formulação a seguir considera grafos *homogêneos* e *estáticos*, uma vez que grafos *heterogêneos* e *dinâmicos* não são escopo destre trabalho.

A principal ideia consiste em integrar módulos GCN e RNN que, conforme a arquitetura, podem operar de forma intercalada ou não. Por exemplo, considere o módulo RNN definido por

$$\mathbf{H}^{(t)} = \sigma\left(\mathbf{W}\mathbf{X}^{(t)} + \mathbf{U}\mathbf{H}^{(t-1)} + \mathbf{b}\right), \tag{II.32}$$

em que $\mathbf{X}^{(t)} \in \mathbb{R}^{N \times F}$ é a matriz de *features* dos vértices no passo de tempo t. Após integrar o módulo GCN, podemos modificar a Equação II.32 para

$$\mathbf{H}^{(t)} = \sigma\left(\mathrm{GCN}(\mathbf{X}^{(t)}, \mathbf{A}; \mathbf{W}) + \mathrm{GCN}(\mathbf{H}^{(t-1)}, \mathbf{A}; \mathbf{U}) + \mathbf{b}\right), \tag{II.33}$$

definindo uma camada da STGCN [Wu et al., 2019].

Capítulo III Trabalhos Relacionados

Para a revisão da literatura empregamos o Portal de Periódicos da CAPES[1] com a seguinte string de busca: *"graph convolution" AND ("time series" OR "spatio-temporal") AND "weather forecasting"*. O portal retorna 26 artigos publicados até junho de 2024, data limite definida para a revisão da literatura em concordância com o prazo de finalização deste trabalho. Foram selecionados para leitura os artigos relacionados ao tema deste trabalho. Durante a leitura, outros artigos foram buscados por meio da aplicação da técnica *SnowBalling*. Por fim, também foram buscados artigos do tipo *survey* em temas como *graph neural networks* e *deep learning-based weather prediction*.

De acordo com os artigos revisados, nota-se um interesse crescente no emprego da arquitetura STGCN para o treinamento de modelos de previsão em diversos domínios que apresentam séries temporais de dados multivariados. No campo da *Astronomia*, Knyazeva et al. [2016] processaram imagens de erupções solares com 5000×5000 *pixels* como um grafo em que o vértice denota valores extremos de erupção e as arestas conexões entre mínimos e máximo locais em uma seção da imagem, viabilizando o treinamento de um modelo de previsão baseado em STGCN. No campo da *Energia*, Verdone et al. [2024] modelaram um grafo em que os vértices representam estações de produção de energia e as arestas a relação entre as estações baseadas em limiares de distância entre elas. Os vértices foram associados à series temporais de valores de energia produzida pela estação e condições meteorológicas na sua localização (*e.g.* temperatura, velocidade e direção do vento).

A maior quantidade de trabalhos empregando STGCNs encontra-se no campo da previsão de tráfego. Em geral, esses trabalhos modelam as malhas de tráfego como um grafo em que os vértices e arestas denotam segmentos interconectados da malha. Os vértices são normalmente associados à condições de trafego em uma determinada localização (*e.g.* velocidade média, quantidade de partidas e chegadas). Dependendo da tarefa de previsão, vértices também costumam ser associados a dados de outros domínios que podem impactar nas condições do trafego, como por exemplo condições do tempo (*e.g.* sol, chuva, neve), características do segmento da malha (*e.g.* largura da rodovia), eventos sociais (*e.g.* feriados, eventos turísticos), dentre outros.

Wang et al. [2020] treinaram modelos de previsão de acidentes de trânsito com dados de vídeos capturados por câmeras internas de veículos automotores e dados meteorológicos, possibilitando alertas de perigo de acidente em tempo real. Zhang et al. [2022] e Zhang et al. [2021] treinaram,

[1] https://www.periodicos.capes.gov.br/

respectivamente, modelos de previsão de atrasos de trem (malha ferroviária) e atrasos em voos (malha aérea) usando dados observados pelas respectivas companhias. Sun et al. [2020] empregaram dados de fluxo de táxis e bicicletas para a previsão de fluxo de pessoas entre sub-regiões da área urbana. Lv et al. [2021]; Hou et al. [2021]; Zhou et al. [2022]; Liu et al. [2022]; Zou et al. [2024] treinaram modelos de previsão de tráfego urbano baseados em STGCN, empregando dados de companhias e serviços de transporte urbano (*e.g.* Uber, táxi, ônibus).

De forma similar, STGCNs são empregadas no treinamento de modelos de previsão para eventos relacionados ao clima. Farahmand et al. [2023] treinaram modelos de previsão de inundação empregando características baseadas na física (*e.g.* intensidade da precipitação e elevação da água) e dados detectados pelo homem (*e.g.* relatos de inundação por moradores). A área considerada foi modelada como um grafo em que vértices denotam sub-regiões da área e as arestas denotam relações entre essas sub-regiões baseadas na distância entre elas e suas características geográficas (*e.g.* elevação). Wang et al. [2024] empregaram dados de estações de monitoramento para previsão da qualidade do ar. Os vértices do grafo também foram associados à dados meteorológicos e características urbanas tais como quantidade de fábricas, residências, serviços de transporte, dentre outros, uma vez que é esperado que tais características tenham relação com a qualidade do ar.

Diversos trabalhos recentes têm buscado empregar ANNs na geração de modelos de previsão do tempo [Ren et al., 2021b]. Em geral, esses trabalhos empregam dados de estações telemétricas, radares ou satélites como base de dados históricos para o treinamento dos modelos. Devido à ausência de estações telemétricas em determinadas áreas, os referidos trabalhos buscam interpolar os dados de forma a inferir valores para áreas não devidamente cobertas.

Liebmann and Allured [2005] empregaram dados de estações telemétricas na construção de um conjunto de dados de precipitação em forma de grade cobrindo a América do Sul. Cada coordenada da grade de dados é interpolada pela média das estações vizinhas dentro de um raio determinado. Funk et al. [2015] combinaram dados de estações telemétricas e de satélites na mesma tarefa, gerando conjuntos de dados com maior resolução espacial (0.05^{o}) e cobertura quase global.

Lewenfus et al. [2020] empregaram uma arquitetura que combina redes neurais recorrentes e processamento de sinais em grafos para interpolar e predizer dados em séries temporais. A arquitetura foi empregada na tarefa de previsão em dados de temperatura do ar oriundos de 430 estações dos Estados Unidos. Uma das aplicações testadas em sua arquitetura consiste no treinamento da rede com um subconjunto $S < N$ dentre N estações telemétricas capaz de predizer valores para todas as N estações. Para tal, empregaram uma função de interpolação sobre os valores preditos pela rede.

Zhou et al. [2021] empregaram dados de direção e velocidade do vento juntamente com medições de estações telemétricas para a previsão de concentração de poluente no ar. Os dados são modelados

em um grafo dirigido onde um vértice representa uma estação telemétrica e uma aresta representa a distância entre duas estações, distância essa calculada com base nas condições do vento. Em seu trabalho, porém, os dados não são convertidos para o formato em grade. O processo de interpolação se limita a predizer dados para os nós do grafo não presentes no conjunto de treinamento.

Tekin et al. [2021] propuseram uma arquitetura híbrida de redes de convolução e redes recorrentes para a previsão de temperatura. Sua arquitetura emprega tanto dados no formato de grade, quanto dados de estações telemétricas. Os dados das estações telemétricas são interpolados para o formato em grade antes de serem passados para o treinamento da rede, o que configura a interpolação como uma etapa de pré-processamento.

Castro et al. [2021] propuseram o *Spatiotemporal Convolutional Sequence to Sequence Network* (STConvS2S) que emprega somente camadas de convolução na predição de séries temporais. Seu trabalho realizou experimentos em conjuntos de dados de precipitação e temperatura do ar, obtendo os melhores resultados em termos de acurácia, consumo de memória e tempo de treinamento. Apesar dos resultados, os conjuntos de dados empregados em seu trabalho foram baixados de fontes que aplicam processos de interpolação similares aos do GPCC e CHC, que em geral apresentam valores discrepantes se comparados aos valores mensurados pelas estações telemétricas.

A Tabela III.1 apresenta uma visão geral das principais características dos trabalhos relacionados à previsão do tempo apresentados neste capítulo. Os trabalhos que empregam a arquitetura *Convolutional Neural Network* (CNN) naturalmente geram previsões em uma grade regular de pontos já que a entrada desse tipo de arquitetura exige esse formato. Por outro lado, quando empregam dados espacialmente não uniformes (*e.g.* estações telemétricas, boias marítimas) acabam tendo que interpolar os dados em uma etapa de pré-processamento.

Já as arquiteturas STGCN podem ser treinadas com dados espacialmente não uniformes, porém a previsão em uma grade regular torna-se uma tarefa mais difícil. Lewenfus et al. [2020] empregaram uma STGCN com dados espacialmente não uniformes e resolveu a questão de previsão em uma grade regular usando uma função de interpolação sobre os valores preditos pela rede. No entanto, a interpolação não faz parte do processo de aprendizado e foi processada somente sobre dados univariados (temperatura).

Tabela III.1: Visão geral dos modelos de previsão.

Trabalho	**Modelo**	**Saída em grade**	**Múltiplas Fontes**	**Multivariado**	**Interpolação por ANN**
Lewenfus et al. [2020]	STGCN	x			
Zhou et al. [2021]	STGCN		x	x	
Castro et al. [2021]	CNN	x		x	
Tekin et al. [2021]	CNN	x	x	x	
Este trabalho	STGCN	x	x	x	x

Os trabalhos relacionados apresentam indícios de que o emprego de múltiplas fontes de dados conduzem a um aprendizado dos padrões da física da atmosfera mais acurado. Gerar previsões em um formato de grade regular exige a interpolação das fontes de dados espacialmente não uniformes. Tal processo de interpolação quando realizado por meio de métodos estatísticos pode conduzir a valores discrepantes o que provavelmente conduzirá a menor acurácia do modelo.

Neste trabalho, a arquitetura é adaptada para que a própria STGCN interpole os dados por meio do aprendizado dos padrões da física da atmosfera em uma série temporal. O modelo é treinado e avaliado empregando dados multivariados de diferentes fontes. Dessa forma, espera-se que o modelo seja capaz de interpolar os dados com maior precisão, maximizando a acurácia na previsão.

Capítulo IV Metodologia

Este capítulo aborda a metodologia empregada neste trabalho. O capítulo é dividido em quatro seções. A Seção IV.1, aborda a formulação do problema considerado nesta dissertação e como o problema é modelado para o aprendizado em uma arquitetura STGCN. Na Seção IV.2, descrevemos as diferentes fontes de dados meteorológicos empregadas neste trabalho. A Seção IV.3 apresenta o *data lake* para armazenamento dos dados meteorológicos e detalha o processo de otimização e curadoria dos dados. Por fim, na Seção IV.4 introduzimos a arquitetura STGCN que é utilizada nos experimentos.

IV.1 Formulação do Problema

Este trabalho tem por objetivo implementar uma STGCN para interpolar dados de precipitação representados como séries temporais de observações meteorológicas. A física da atmosfera é considerada um sistema caótico e portanto é intuitivo pensar que modelar tal sistema depende do maior número possível de dados sobre ele. Devido a esse cenário, a STGCN é modelada para treinar com dados de distintos instrumentos de observação meteorológica (*e.g.* radar, estações telemétricas).

O problema pode ser formulado como uma tarefa de aprendizado em grafos heterogêneos e dinâmicos. Tal afirmação leva em conta que dados de diferentes instrumentos meteorológicos possuem diferentes variáveis associadas. Além disso, determinados instrumentos podem parar de funcionar por um período de tempo, o que altera a topologia de um grafo associado. No entanto, para fins de simplificação, este trabalho modela o problema como um aprendizado em grafos homogêneos e estáticos. O Capítulo V discursa sobre o pré-processamento realizado para atender a referida simplificação.

Alguns instrumentos meteorológicos registram observações em maior alcance e resolução espacial, enquanto outros são mais limitados. Por exemplo, radares meteorológicos têm um alcance de varredura acima de 200 km a partir de sua localização e são capazes de identificar hidrometeoros na atmosfera em alta resolução espacial, em geral em torno de 500 metros. Já as estações telemétricas registram as variáveis meteorológicas observadas (*e.g.* precipitação, velocidade do vento) somente para a coordenada geográfica em que estão localizadas (latitude, longitude).

A distribuição espacial não uniforme das estações telemétricas conduz à necessidade de inter-

polar os dados de precipitação para obter uma estimativa sobre as regiões não cobertas. Uma das formas de processar a interpolação é definir previamente uma grade regular de coordenadas geográficas sobre a região de interesse. Para cada coordenada da grade, aplica-se um método de interpolação com base nos dados de precipitação observados pelas estações telemétricas próximas à coordenada. Com a grade regular de dados de precipitação, pode-se treinar uma ANN para predizer os valores de precipitação futuros para toda a área de interesse. O ponto a ser ressaltado é que uma interpolação imprecisa conduzirá à uma baixa qualidade de dados de treinamento para a ANN.

A Figura IV.1 apresenta uma visão geral da arquitetura proposta neste trabalho. Os dados coletados pelos instrumentos meteorológicos são introduzidos na rede na forma de séries temporais de grafos. A STGCN aprende a interpolar os dados empregando séries temporais multivariadas de múltiplos instrumentos meteorológicos. Os dados interpolados podem ser passados como entrada para outro tipo de ANN para processar a tarefa de previsão. Na Figura IV.1, o módulo de ANN envolto pelo retângulo tracejado não é escopo deste trabalho. No entanto, a arquitetura completa pode ser perfeitamente definida, empilhando os distintos módulos em uma única rede de aprendizado.

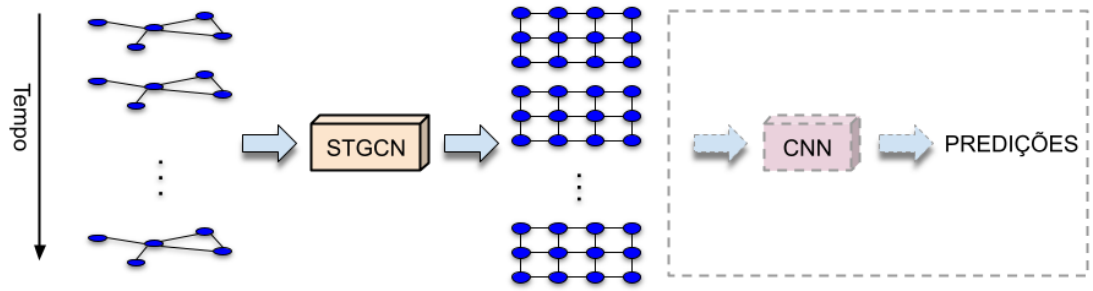


Figura IV.1: Visão geral da arquitetura. As séries temporais de grafos são passadas como entrada para a STGCN que interpola os dados. Os dados interpolados podem ser passados para outras camadas de ANN que processam a previsão. O módulo de ANN envolto pelo retângulo tracejado não é escopo deste trabalho.

Neste trabalho, o problema é formulado como se segue. A Figura I.1 apresenta os limites da área territorial de estudo da cidade do Rio de Janeiro. Inicialmente a área territorial é mapeada como uma grade de pontos $\mathbf{\Gamma}$ uniformemente distribuídos, de forma que cada ponto $\gamma \in \mathbf{\Gamma}$ represente uma coordenada geográfica. O conjunto de coordenadas $\mathbf{P}$ de estações telemétricas é adicionado à grade $\mathbf{\Gamma}$, gerando a grade irregular $\mathbf{\Upsilon} = \mathbf{\Gamma} \cup \mathbf{P}$.

O conjunto de coordenadas $v \in \mathbf{\Upsilon}$ representam os vértices de um grafo. Cada vértice é vinculado a um vetor $\mathbf{x} \in \mathbb{R}^F$ em que F representa o número de variáveis meteorológicas. As arestas são definidas por um limiar de distância q entre as coordenadas, aplicando a distância *Haversine*.

Empregando os dados meteorológicos e a grade irregular $\boldsymbol{\Upsilon}$, são processadas séries temporais de grafos $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_T\}$. $\mathcal{G}_t = (\mathbf{A}, \mathbf{X}_t)$ representa um grafo no passo de tempo $t \in \{1, 2, ..., T\}$. $\mathbf{X} \in \mathbb{R}^{N \times F}$ representa a matriz de *features* (variáveis meteorológicas) associadas aos vértices. O modelo é treinado para aproximar uma função $f(\cdot)$ que mapeia uma sequência de variáveis meteorológicas observadas para uma grade de valores de precipitação interpolados. A função $f(\cdot)$ pode ser formalmente definida pela Equação IV.1. Nessa equação, $\mathcal{Z}$ representa a matriz de valores de precipitação interpolados. Considerando que $\mathcal{Z}$ é uma grade irregular de valores interpolados conforme $\boldsymbol{\Upsilon}$, podemos obter a grade regular de valores interpolados processando $\boldsymbol{\Upsilon} \cap \boldsymbol{\Gamma}$.

$$\mathcal{Z}_t = f(\mathbf{X}_{t-T}, \cdots, \mathbf{X}_{t-1}, \mathbf{X}_t), \tag{IV.1}$$

IV.2 Fontes de Dados

A primeira etapa deste trabalho consiste na coleta e pré-processamento dos dados necessários para o experimento. A cidade do Rio de Janeiro conta com diversas fontes de dados meteorológicos coletados por instrumentos operados por diferentes instituições. Dentre os instrumentos podem-se citar: satélites, radares meteorológicos, estações pluviométricas, estações meteorológicas, radiossonda, boias marítimas. Neste trabalho são empregadas as seguintes fontes de dados, todas com seus instrumentos de observação instalados dentro da área de estudo:

- 33 estações telemétricas do sistema Alerta Rio.
- 4 estações telemétricas do INMET.
- 1 radar meteorológico do INEA.

Dentre as 33 estações telemétricas do Alerta Rio 25 são somente pluviométricas, 6 são meteorológicas parciais e 2 são meteorológicas completas. As 4 estações do INMET são meteorológicas completas. As características e variáveis de cada instrumento serão melhor detalhados na Seção V.1.

A Figura IV.2 apresenta a localização dos instrumentos supracitados. Os pequenos pontos verdes representam os pluviômetros do Alerta Rio. Os ícones de informação nas cores verde, laranja e azul representam, respectivamente, as estações meteorológicas do Alerta Rio, as estações meteorológicas do INMET e o radar meteorológico do INEA.

IV.3 Data Lake

Os arquivos de dados meteorológicos coletados se apresentam em diferentes formatos, tais como *json*, *hdf* e *txt*. Armazenar tal diversidade de formatos conduz à necessidade de uma tecnologia de armazenamento de objetos. Por conta disso, os dados coletados neste trabalho são armazenados

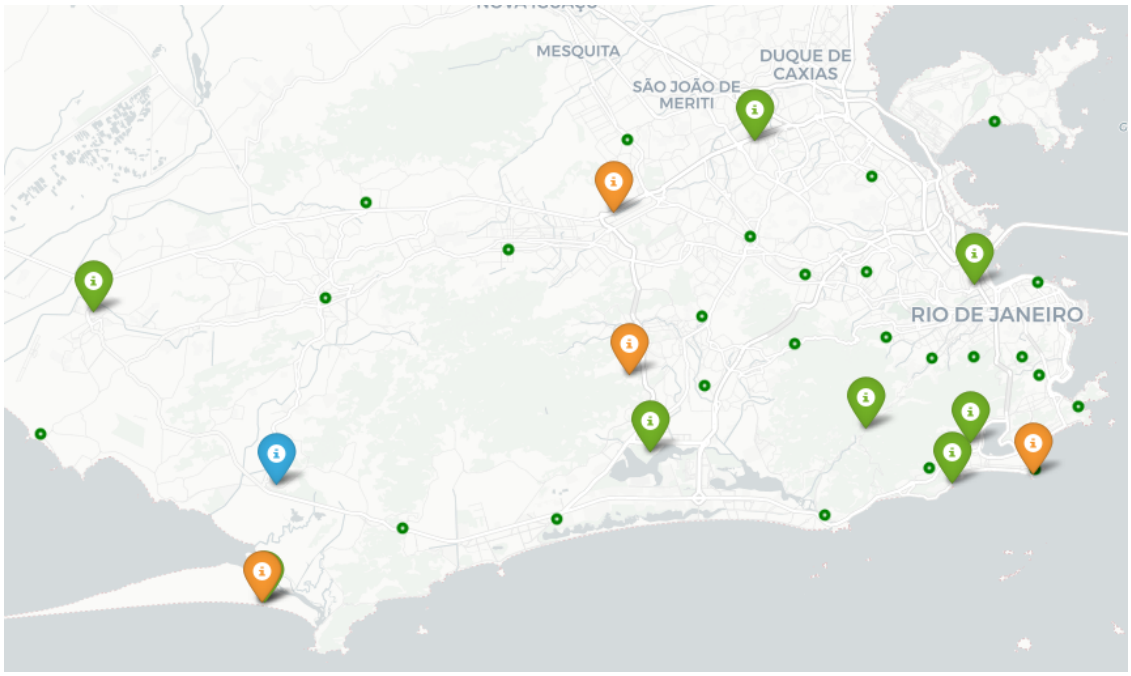


Figura IV.2: Localização dos instrumentos de observação. Pontos em verde representam os pluviômetros do Alerta Rio. Os ícones de informação nas cores verde, laranja e azul representam, respectivamente, as estações meteorológicas do Alerta Rio, as estações meteorológicas do INMET e o radar meteorológico do INEA.

em uma instância de *data lake*, empregando a tecnologia *MinIO*[1] que é *open source*. Além disso, é definido um processo de otimização e curadoria dos dados com a finalidade de viabilizar a análise exploratória e o pré-processamento, bem como garantir a qualidade dos dados.

Os dados em um *data lake* são organizados em *buckets*. *Bucket* é um endereço raiz do *data lake* e remete à um recipiente em que agrupamos objetos baseado em algum atributo. Um *bucket* pode ser dividido em vários subníveis, de forma a organizar os objetos. A Figura IV.3 apresenta a estrutura de *buckets* definida para o *data lake* deste trabalho, bem como as etapas do processo de otimização e curadoria. Conforme apresenta a Figura IV.3, são definidos três *buckets*: *landing*, *staged* e *curated*. Todos os *buckets* são subdivididos por instrumento meteorológico e instituição. Além disso, todos os *scripts* de processamento dos dados são implementados separadamente para cada instituição/instrumento, uma vez que cada fonte de dados possui características distintas.

O *bucket landing* é o responsável por armazenar os dados brutos, *i.e.*, os dados em seu formato original. Quando o volume de dados original é muito grande, é recomendado aplicar algum filtro diretamente na coleta dos dados. Apesar disso, recomenda-se que tal transformação realize o mínimo de alteração na estrutura original dos dados. Por exemplo, em uma fonte de dados tabular podemos filtrar determinadas colunas de dados, o que preserva a estrutura das demais colunas coletadas e armazenadas.

A primeira etapa do processo consiste em otimizar os dados do *bucket landing*. O *script* de

[1] https://min.io/

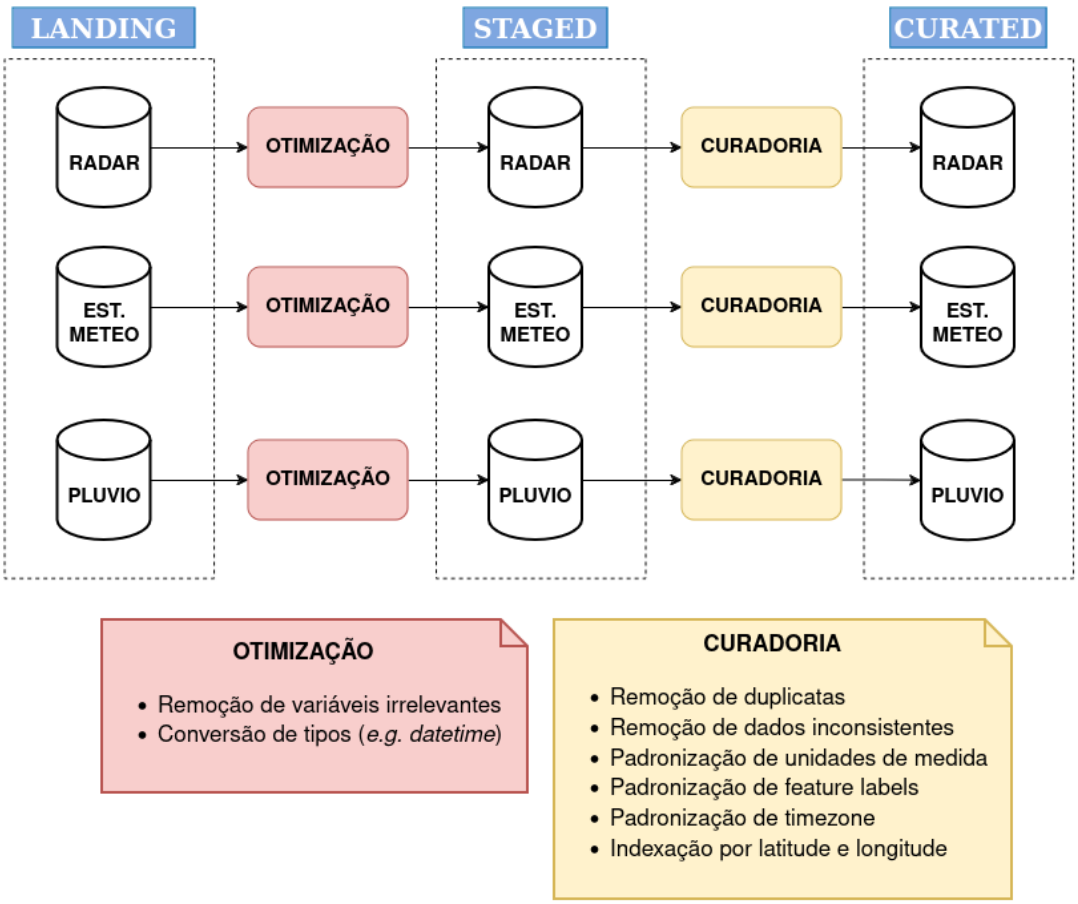


Figura IV.3: *Data Lake*. Os dados meteorológicos coletados se apresentam em diferentes estruturas tais como *json*, *hdf* e *txt*. Os dados coletados são armazenados no *bucket landing* do *data lake*. Em seguida passam pelos processos de *otimização* e *curadoria*, conforme detalhado na imagem. Por fim, os dados são armazenados no *bucket curated* no formato *parquet*.

otimização dos dados tem três finalidades: *i*) eliminar dados irrelevantes; *ii*) converter os tipos de dados (*e.g. datetime, string, numeric*) e; *iii*) armazenar os dados transformados no formato *parquet*[2]. Os dados otimizados são armazenados no *bucket staged*.

A segunda etapa consiste na curadoria dos dados. Esta etapa tem duas finalidades de mais alto nível: *i*) aplicar técnicas comuns empregadas no pré-processamento de dados para AM e; *ii*) garantir a qualidade dos dados. Um detalhe importante é que todas as transformações executadas nesta etapa devem ser independentes de integração das fontes de dados.

As principais transformações de curadoria aplicadas neste trabalho são: *i)* remoção de duplicatas; *ii)* remoção de dados inconsistentes; *iii)* padronização de unidades de medida; *iv)* padronização de nomenclatura de variáveis; *v)* padronização do *timezone* e; *vi)* indexação por latitude e longitude. Os dados processados nesta etapa são armazenados no *bucket curated*, também no formato *parquet*. As operações de padronização de unidades de medida e nomenclatura de variáveis são particularmente importantes para viabilizar a integração dos dados. A partir deste ponto, os dados

[2]https://parquet.apache.org/

armazenados no *bucket curated* ficam disponíveis para a integração. Uma vez que os dados foram processados separadamente por instituição/instrumento, diferentes integrações podem ser processadas por meio da seleção de subconjuntos destes.

IV.4 Arquitetura STGCN

A arquitetura STGCN empregada neste trabalho é a proposta por Bai et al. [2021]. A arquitetura implementa o aprendizado de representações em séries temporais de grafos, além de aplicar um mecanismo de *soft-attention* que captura o contexto global da série temporal para melhorar a acurácia das predições [Bai et al., 2021]. Além disso, sua implementação encontra-se disponível no *framework PyTorch Geometric Temporal* [Rozemberczki et al., 2021].

A Figura IV.4 apresenta a visão geral da arquitetura STGCN proposta por Bai et al. [2021]. O módulo T-GCN é composto por uma GCN e uma GRU que processam as convoluções espaço-temporais. A GCN empregada é a proposta por Kipf and Welling [2016]. O módulo recebe como entrada uma série temporal com n passos de tempo e gera na saída n *hidden states* $\{\mathbf{h}_{t-n}, \cdots, \mathbf{h}_{t-1}, \mathbf{h}_t\}$. O cálculo processado pelo módulo T-GCN é apresentado nas Equações IV.2, IV.3, IV.4, IV.5.

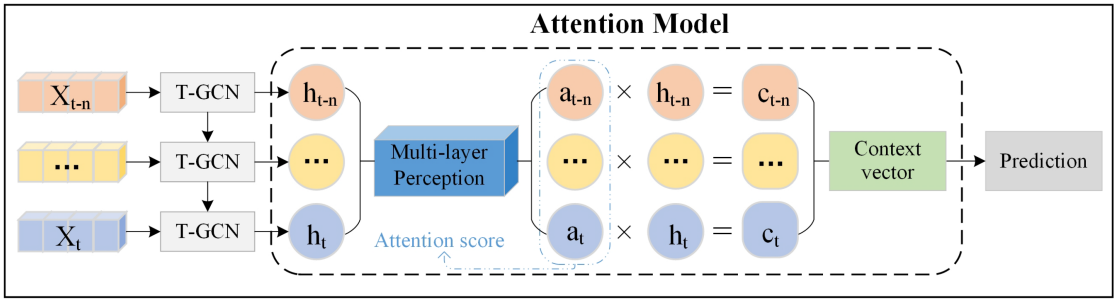


Figura IV.4: Visão geral da arquitetura STGCN proposta por Bai et al. [2021]. Imagem: Bai et al. [2021].

$$\mathbf{u}_t = \sigma(\mathbf{W}_u * [\mathrm{GCN}(\mathbf{A}, \mathbf{X_t}), \mathbf{h}_{t-1}] + \mathbf{b}_u) \tag{IV.2}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r * [\mathrm{GCN}(\mathbf{A}, \mathbf{X_t}), \mathbf{h}_{t-1}] + \mathbf{b}_r) \tag{IV.3}$$

$$\mathbf{c}_t = \tanh(\mathbf{W}_c * [\mathrm{GCN}(\mathbf{A}, \mathbf{X_t}), (\mathbf{r}_t * \mathbf{h}_{t-1})] + \mathbf{b}_c) \tag{IV.4}$$

$$\mathbf{h}_t = \mathbf{u}_t * \mathbf{h}_{t-1} + (1 - \mathbf{u}_t) * \mathbf{c}_t \tag{IV.5}$$

As *hidden states* são introduzidas em um modelo *soft-attention* de forma a determinar o *vetor de contexto* que captura a informação global. O modelo emprega uma MLP e uma *softmax* para calcular os pesos de atenção $\boldsymbol{\alpha}$, definidas nas Equações IV.6 e IV.7.

$$\mathbf{e}_i = \mathbf{w}_{(2)}(\mathbf{w}_{(1)}\mathbf{H} + \mathbf{b}_{(1)}) + \mathbf{b}_{(2)} \tag{IV.6}$$

$$\boldsymbol{\alpha}_i = \frac{\exp(\mathbf{e}_i)}{\sum_{k=1}^{n} \exp(\mathbf{e}_k)}. \tag{IV.7}$$

O vetor de contexto é calculado pela média ponderada pelos pesos $\boldsymbol{\alpha}$ conforme definido na Equação IV.8

$$\mathbf{C}_t = \sum_{i=1}^{n} \boldsymbol{\alpha_i} * \mathbf{h}_i. \tag{IV.8}$$

Por fim, o vetor de contexto é passado para camadas MLP que processam a tarefa de previsão.

Capítulo V Avaliação Experimental

Este Capítulo está organizado como segue. A Seção V.1 apresenta uma análise exploratória dos dados empregados. A Seção V.2 descreve o procedimento de construção do grafo a partir dos dados disponíveis. A Seção V.3 descreve atividades relacionadas à otimização e curadoria, pré-processamento e integração dos dados, além de detalhar a construção das séries temporais de grafos. A Seção V.4 apresenta detalhes relativos ao procedimento de treinamento e avaliação da STGCN.

V.1 Análise Exploratória dos Dados

Esta seção detalha a análise exploratória realizada sobre os dados meteorológicos empregados neste trabalho. Cada subseção detalha uma fonte de dados distinta, descrevendo as variáveis meteorológicas disponíveis para a fonte e a análise exploratória específica. O tratamento aplicado a problemas identificados (*e.g.* dados faltantes, dados espúrios) é detalhado na Seção V.3.

Como citado na Seção IV.2 foram coletados os dados do radar meteorológico do INEA e das estações telemétricas do INMET e Alerta Rio. Devido ao grande volume de dados gerado pelo radar do INEA e por não existir uma API disponível para *download*, foi necessário obter os dados por meio de um HD externo diretamente com a instituição. Devido a isso, somente um subconjunto de datas pôde ser copiado.

Para estabelecer as datas de dados de radar a serem usadas, foi empregada a classificação dos níveis de precipitação definida pelo Alerta Rio e apresentada na Tabela V.1. Por meio do processamento dos dados dos pluviômetros do sistema Alerta Rio, foram identificadas todas as datas em que ocorreu *Chuva forte* a partir da data de início de operação do radar do INEA. Denotamos por $\mathcal{D}'$ as datas identificadas. Para cada data em $\mathcal{D}'$ identificamos a data imediatamente anterior gerando o conjunto final de datas $\mathcal{D} = \mathcal{D}' \cup \{(\forall d' \in \mathcal{D}')d' - 1\}$.

É esperado que instrumentos meteorológicos sofram interrupções de operação por curtos ou longos períodos e/ou que os dados registrados possam conter erros devido a falhas de hardware ou software. Devido a tal possibilidade, foi realizada uma análise exploratória em cada fonte de dados buscando avaliar a qualidade dos dados. Foram observadas as seguintes questões: *i)* duplicatas temporais, *i.e.*, dois ou mais registros para o mesmo horário com valores distintos observados; *ii)*

Tabela V.1: Classificação dos níveis de precipitação definida pelo Alerta Rio.

Classificação	**Nível de precipitação (ρ) em mm/h**
Sem chuva	$\rho = 0.0$
Chuva fraca	$0.0 < \rho < 5.0$
Chuva moderada	$5.0 \leq \rho < 25.0$
Chuva forte	$25.0 \leq \rho < 50.0$
Chuva muito forte	$50 \leq \rho$

período de operação da estação e quantidade de registros esperada por intervalo de tempo; *iii)* dados faltantes; *iv)* dados espúrios; *v)* análise da distribuição das variáveis observadas e; *vi)* outras questões de qualidade encontradas durante a análise.

V.1.1 Radar Meteorológico do INEA

O radar meteorológico do INEA está instalado no bairro de Guaratiba na cidade do Rio de Janeiro, nas coordenadas aproximadas (latitude: -22.993, longitude: -43.588) (veja Figura IV.2). O radar iniciou sua operação em 2015 e registra os hidrometeoros detectados em um raio de até 250 km a partir de sua posição. A cada 5 min o radar opera uma seção de escaneamento em 360 graus de azimutes e 8 ângulos de elevação. A Figura V.1 apresenta a estratégia de escaneamento do radar do INEA.

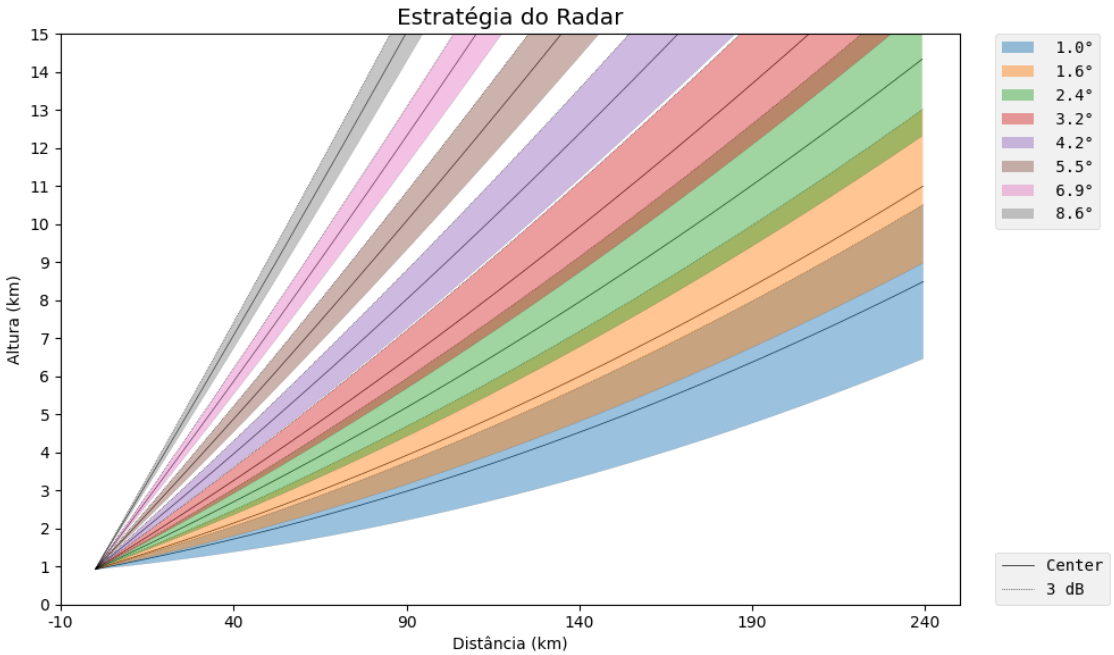


Figura V.1: Estratégia de escaneamento do radar do INEA. O radar do INEA escaneia 8 ângulos de elevação, entre 1,0 e 8,6 graus.

Os arquivos disponibilizados pelo INEA para serem empregados neste trabalho estão no formato hierárquico HDF[1]. Cada arquivo representa uma seção completa de escaneamento do radar que é

[1] *Hierarchical Data Format* (HDF) é um formato de arquivo de dados projetado pelo *National Center for Super-*

operada a cada 5 minutos. Em média, cada dia completo de dados possui 288 arquivos HDF, totalizado aproximadamente 1,3 GB de dados diários. Os arquivos são estruturados no padrão OPERA5[2] e foram transformados para um formato tabular com as seguintes colunas de dados: data/hora da observação, latitude, longitude, altitude e refletividade observada.

Dentre as datas solicitadas ao INEA foram disponibilizadas 224 datas entre 2016 e 2023. Apesar de existirem 11 parâmetros mensurados pelo radar, apenas o parâmetro DBZ foi empregado neste trabalho. Tal decisão é apoiada por meio da consultoria de meteorologistas do Alerta Rio que indicam ser o parâmetro mais relevante. O parâmetro DBZ refere-se à *refletividade horizontal* mensurada pelo radar o qual registra os valores em unidades de dBZ (*Decibel relative to Z*). Devido a alta resolução espacial (500 metros) e temporal (5 minutos) dos dados, foi processada a agregação da variável *refletividade* pela média para uma resolução espacial de 1 km (latitude, longitude e altitude) e temporal de 1 hora antes de proceder a análise exploratória.

Dentre os registros, não foram encontradas duplicatas temporais. A Figura V.2a apresenta a contagem de registros para cada data. É possível perceber dois platôs que representam a contagem esperada de registros para cada sub-período. Um posicionado próximo a 1.0×10^6 e outro próximo a 0.4×10^6. Além disso, alguns pontos isolados abaixo desses dois platôs denotam datas com quantidade de registros abaixo do esperado, provavelmente devido à falha de *hardware* ou *software*. Os platôs se justificam devido a uma mudança na operação do radar em dezembro de 2017 que passou a escanear menos ângulos de elevação e, portanto, passou a registrar uma quantidade menor de observações. Por consequência dessa mudança a altitude máxima observada passou a ser de 8 km. A Figura V.2b apresenta as altitudes máximas observadas para cada ano.

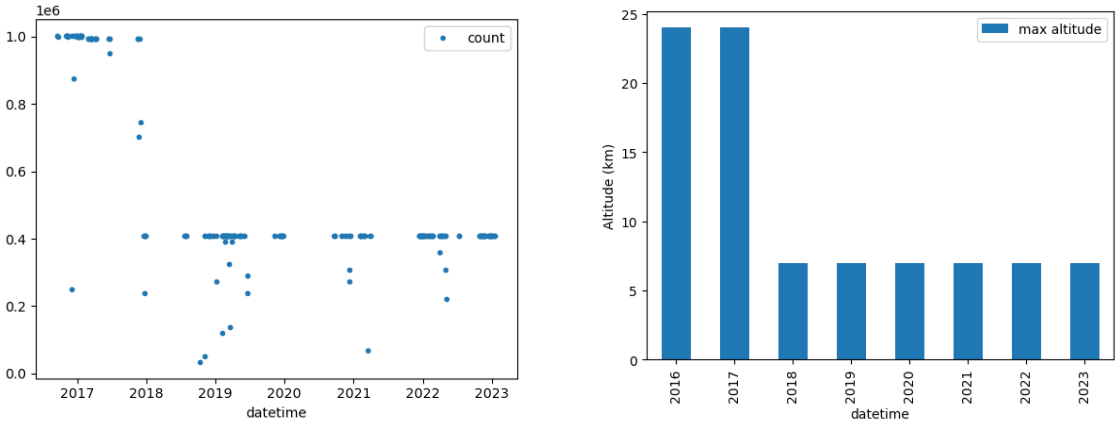
O efeito do *cone do silêncio*, detalhado na Seção II.1, foi confirmado durante a análise dos registros por altitude. A quantidade de registros associados à altitudes maiores é significativamente menor do que para altitudes menores. A Figura V.3 apresenta os dados do radar do INEA para uma data/hora arbitrária com os pontos observados nos quatro primeiros níveis de altitude. Conforme aumenta o nível de altitude, aumenta também a área não observada em torno do radar.

Não foram encontrados dados faltantes nem dados espúrios. Quanto à distribuição do valor de refletividade percebeu-se médias máximas em torno de 150 dBZ para altitudes até 8 km. A partir de 9 km as médias máximas caem significativamente, ficando em torno de 12 dBZ a partir de 14 km.

---

computing Applications (NCSA) otimizado para o armazenamento e manipulação de dados científicos. Atualmente o formato é mantido pelo *HDF Group* (https://www.hdfgroup.org/)

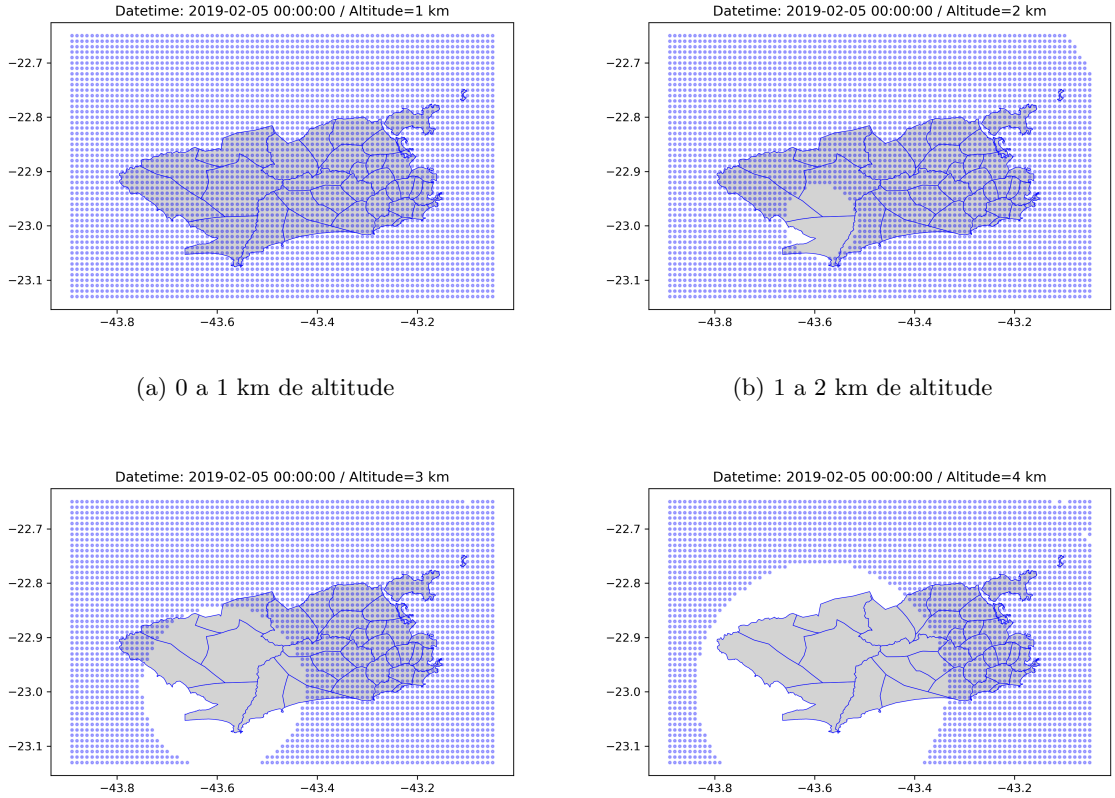
[2]OPERA é um modelo de informação estabelecido pela *European Meteorological Network* (EUMETNET) disponível em https://www.eumetnet.eu/wp-content/uploads/2019/05/OPERA-ODIM_H5-v2.01.pdf



(a) quantidade de registros por data

(b) altitudes máximas observadas por ano

Figura V.2: (a) O gráfico apresenta a quantidade de registros para cada data. Destacam-se dois períodos, um próximo a 1.0×10^6 e outro próximo a 0.4×10^6. A diminuição da quantidade de registros entre os períodos ocorreu devido à mudança de operação do radar que passou a escanear menos ângulos de elevação. (b) Em dezembro de 2017 o radar passou a escanear menos ângulos de elevação o que reduziu a altitude máxima observada.



(a) 0 a 1 km de altitude

(b) 1 a 2 km de altitude

(c) 2 a 3 km de altitude

(d) 3 a 4 km de altitude

Figura V.3: Efeito do cone do silêncio na área de estudo. Pontos em azul denotam uma coordenada observada pelo radar. Conforme aumenta o nível de altitude, menos pontos em torno do radar são observados.

V.1.2 Estações Meteorológicas do INMET

Dentro dos limites da área de estudo deste trabalho, existem quatro estações meteorológicas do INMET. Suas localizações podem ser visualizadas na Figura IV.2. As coordenadas aproximadas e o ano de início de operação das estações são apresentados na Tabela V.2.

Tabela V.2: Coordenadas e período de operação das estações do INMET

Estação	**Latitude / Longitude**	**Ano de Início de Operação**
Marambaia	[-23.050, -43.596]	2002
Forte de Copacabana	[-22.988, -43.191]	2007
Vila Militar	[-22.861, -43.411]	2007
Jacarepaguá	[-22.940, -43.403]	2017

A resolução temporal dos dados é de 1 hora. Dentre as variáveis observadas pelas estações, a Tabela V.3 apresenta as selecionadas para este trabalho. Ressalta-se que o nome de cada variável não necessariamente é igual ao nome empregado no arquivo de dados original. A renomeação das variáveis foi citada na Seção IV.3 e tem por finalidade identificar uma mesma variável meteorológica em diferentes fontes de dados.

Tabela V.3: Seleção das variáveis observadas pelas estações do INMET

Variável	**Descrição**	**Unidade**
pressure	Pressão atmosférica instantânea	mB
temperature	Temperatura instantânea	°C
*dew*_point	Temperatura do ponto de orvalho	°C
humidity	Umidade relativa do ar	%
*wind*_dir	Direção horária do vento	°(gr)
*wind*_speed	Velocidade do vento	m/s
precipitation	Precipitação acumulada	mm

Não foram encontradas duplicatas temporais nos dados, nem contagem inconsistente de registros por data. A Figura V.4 apresenta a análise de dados faltantes por estação/variável meteorológica. O percentual de dados faltantes varia entre 5% e 20% na maioria dos casos. As estações *Vila Militar* e *Forte de Copacabana* são as que apresentam menor percentual de dados faltantes, em torno de 5% na maioria dos casos.

No arquivo original de dados sempre que a variável *wind_speed* registra o valor 0.0 (*i.e.*, ausência de vento) a variável *wind_dir* é registrada como nula. O valor nulo nesse caso não deve ser considerado como dado faltante. Por conta desse caso especial, foram computados os valores nulos reais para a variável *wind_dir*, *i.e.*, *wind_dir* nulo quando *wind_speed* é diferente de 0.0. Somente a estação *Forte de Copacabana* apresentou *wind_dir* nulo verdadeiro em um percentual de 5.91%

Na maioria dos casos não foram encontrados dados espúrios. A exceção fica para a variável meteorológica *dew_point*. A Figura V.5 apresenta a distribuição dessa variável para as quatro

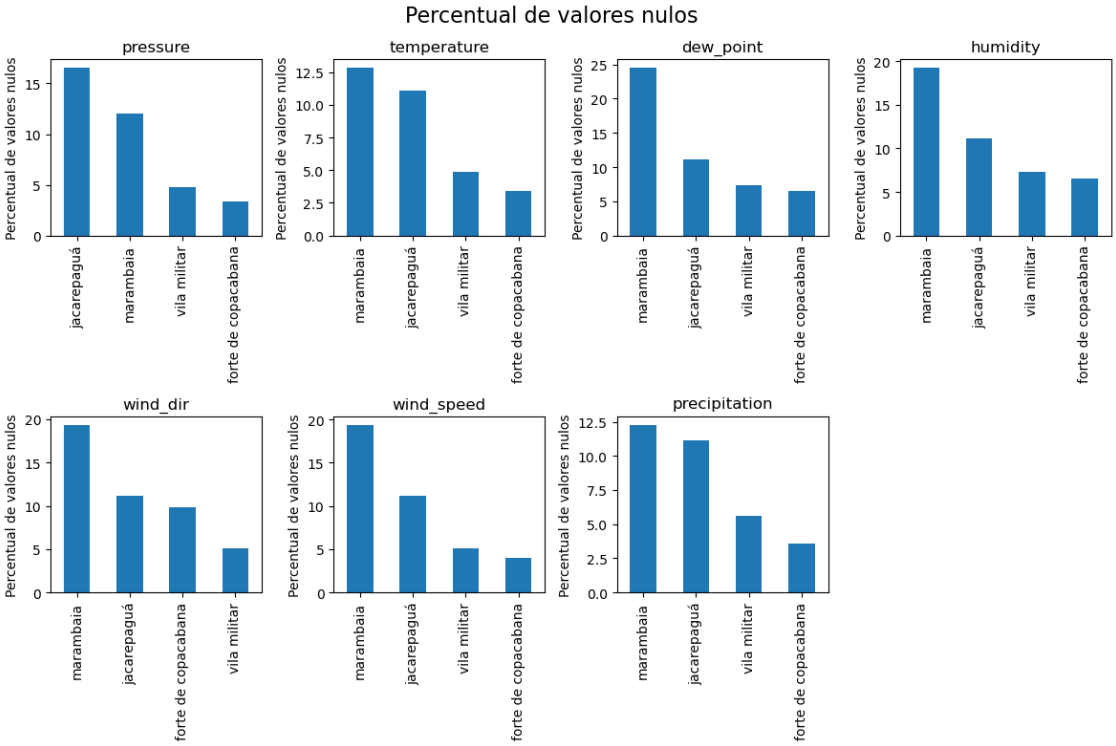


Figura V.4: Percentual de valores nulos das variáveis meteorológicas observadas pelas estações do INMET.

estações. O gráfico apresenta valores notoriamente fora dos limites padrões da distribuição histórica dos dados. Em alguns casos, registros de valores negativos.

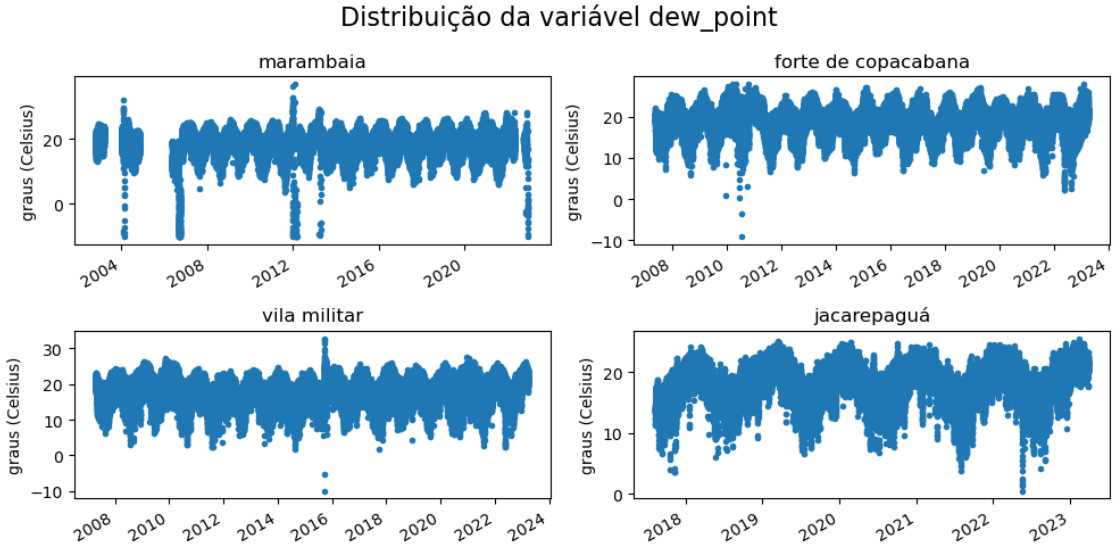


Figura V.5: Distribuição da variável *dew_point* para as estações do INMET. Alguns valores estão fora dos limites padrões da distribuição histórica dos dados, em alguns casos apresentando valores negativos.

V.1.3 Estações Meteorológicas do Alerta Rio

Dentro dos limites da área de estudo deste trabalho existem oito estações meteorológicas do Alerta Rio. Suas localizações podem ser visualizadas na Figura IV.2. As coordenadas aproximadas e o ano de início de operação das estações são apresentados na Tabela V.4. A Tabela V.5 apresenta as variáveis meteorológicas observadas por cada estação. As estações *Sao Cristovao* e *Guaratiba* são consideradas *meteorológicas completas* já que observam todo o conjunto de variáveis. As demais são consideradas *meteorológicas parciais*.

Tabela V.4: Coordenadas e período de operação das estações meteorológicas do Alerta Rio

Estação	**Latitude / Longitude**	**Ano de Início de Operação**
Sao Cristovao	[-22.897, -43.222]	2000
Guaratiba	[-23.050, -43.595]	2004
Alto da Boa Vista	[-22.966, -43.278]	2011
Iraja	[-22.827, -43.337]	2014
Jardim Botanico	[-22.973, -43.224]	2014
Riocentro	[-22.977, -43.392]	2014
Santa Cruz	[-22.909, -43.684]	2014
Vidigal	[-22.992, -43.233]	2020

Tabela V.5: Variáveis observadas pelas estações meteorológicas do Alerta Rio

Estação	*precipitation*	*temperature*	*humidity*	*wind (dir/speed)*	*pressure*
Sao Cristovao	X	X	X	X	X
Guaratiba	X	X	X	X	X
Iraja	X	X	X		
Jardim Botanico	X	X	X		
Riocentro	X	X	X		
Santa Cruz	X	X	X		
Alto da Boa Vista	X	X	X		
Vidigal	X			X	

A resolução temporal dos dados é de 15 minutos. Ressalta-se que o nome de cada variável não necessariamente é igual ao nome empregado no arquivo de dados original. A renomeação das variáveis foi citada na Seção IV.3 e tem por finalidade identificar uma mesma variável meteorológica em diferentes fontes de dados.

Esta seção não apresenta a análise exploratória da variável *precipitation*. Essa variável é analisada na Seção V.1.4 juntamente com os dados das estações pluviométricas. A análise exploratória se limita ao período de dados entre 2016 e 2023, período considerado neste trabalho e que alinha com o ano de início de operação do radar meteorológico do INEA.

Não foram encontradas duplicatas temporais nos dados. Os anos de 2020 e 2021 apresentaram contagem de registros por data superior ao esperado. No entanto, confirmamos que os registros

excedentes ocorrem por conta de um período em que as estações passaram a registrar observações em uma resolução temporal maior (5 minutos).

A Figura V.6 apresenta a análise de dados faltantes por estação/variável meteorológica. O percentual de valores nulos para a variável *wind_dir* deve desconsiderar o caso em que *wind_speed* é 0.0, conforme explicado na Seção V.1.2. Quando filtramos esses casos, as variáveis *wind_dir* e *wind_speed* apresentam quantidade de valores nulos abaixo de 5%. As variáveis *temperature* e *humidity* variam aproximadamente entre 0.0 e 30%, enquanto a variável *pressure* é quase completa, *i.e.*, abaixo de 0.06% de valores nulos.

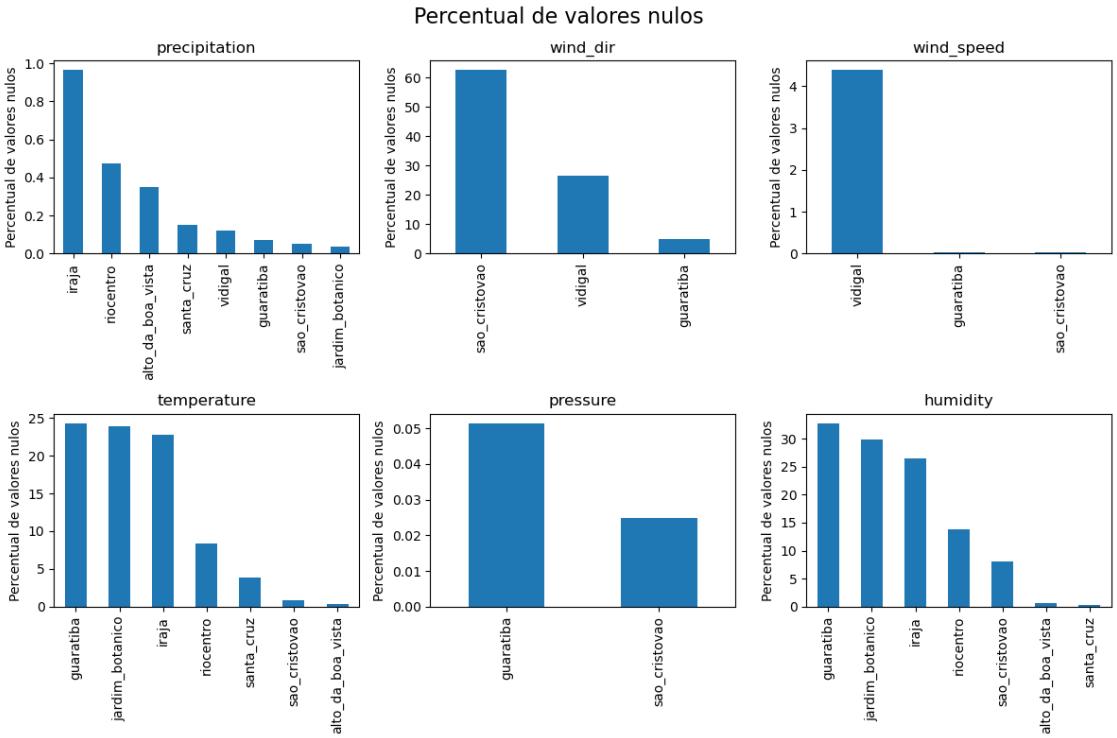


Figura V.6: Percentual de valores nulos das variáveis observadas pelas estações meteorológicas do Alerta Rio.

Os casos mais relevantes de dados espúrios durante a análise exploratória foram identificados nas variáveis *wind_speed* e *temperature*. Na Figura V.7 observamos um ponto extremo para *wind_speed* de aproximadamente 150 km/h registrado pela estação *Vidigal*, valor esse considerado pelos meteorologistas do Alerta Rio como espúrio. Na Figura V.8 observamos alguns casos da variável *temperature* com valores negativos.

V.1.4 Estações Pluviométricas do Alerta Rio

Dentro dos limites da área de estudo deste trabalho existem 25 estações pluviométricas do Alerta Rio. Suas localizações podem ser visualizadas na Figura IV.2. As coordenadas aproximadas são apresentadas na Tabela V.6. Todas as estações pluviométricas já se encontravam em operação em 2016, ano de início de operação do radar meteorológico do INEA. Por conta disso, omitimos

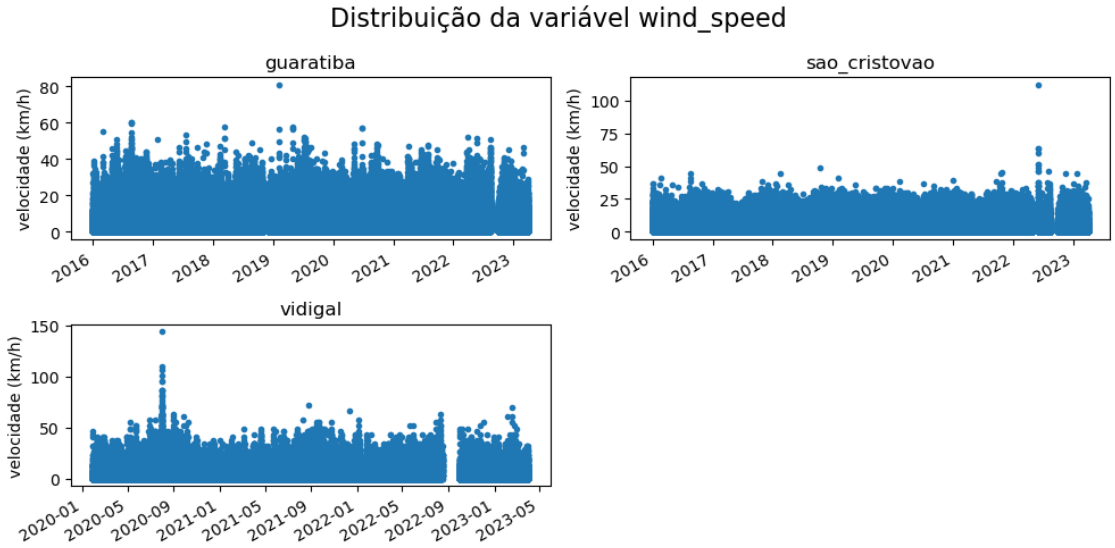


Figura V.7: Distribuição da variável *wind_speed* para as estações meteorológicas do Alerta Rio. Alguns valores estão fora dos limites padrões da distribuição histórica dos dados.

esse dado na Tabela V.6. A análise exploratória apresentada nesta seção também considera a precipitação observada pelas estações meteorológicas do Alerta Rio listadas na Tabela V.4.

Tabela V.6: Coordenadas das estações pluviométricas do Alerta Rio

Estação	**Latitude / Longitude**	**Estação**	**Latitude / Longitude**
Anchieta	[-22.827, -43.403]	Madureira	[-22.873, -43.339]
Av brasil mendanha	[-22.857, -43.541]	Penha	[-22.844, -43.275]
Bangu	[-22.880, -43.466]	Piedade	[-22.892, -43.310]
Barrinha	[-23.008, -43.300]	Recreio	[-23.010, -43.441]
Campo grande	[-22.904, -43.562]	Rocinha	[-22.986, -43.245]
Cidade de deus	[-22.946, -43.363]	Santa teresa	[-22.932, -43.196]
Copacabana	[-22.986, -43.189]	Saude	[-22.896, -43.188]
Grajau	[-22.922, -43.267]	Sepetiba	[-22.969, -43.712]
Grajau jacarepagua	[-22.926, -43.316]	Tanque	[-22.913, -43.365]
Grande meier	[-22.891, -43.278]	Tijuca	[-22.932, -43.222]
Grota funda	[-23.014, -43.521]	Tijuca muda	[-22.933, -43.243]
Ilha do governador	[-22.818, -43.210]	Urca	[-22.956, -43.167]
Laranjeiras	[-22.941, -43.188]		

A resolução temporal dos dados é de 15 min. O nome da variável não necessariamente é igual ao nome empregado no arquivo de dados original. A renomeação da variável foi citada na Seção IV.3 e tem por finalidade identificar uma mesma variável meteorológica em diferentes fontes de dados. A análise exploratória se limita ao período de dados entre 2016 e 2023, período considerado neste trabalho e que alinha com o ano de início de operação do radar meteorológico do INEA.

Não foram encontradas duplicatas temporais nos dados, nem contagem inconsistente de registros por data. A Figura V.9 apresenta a análise de dados faltantes da variável *precipitation* por estação. Em todos os casos, o percentual fica abaixo de $1,1\%$ o que é especialmente interessante neste

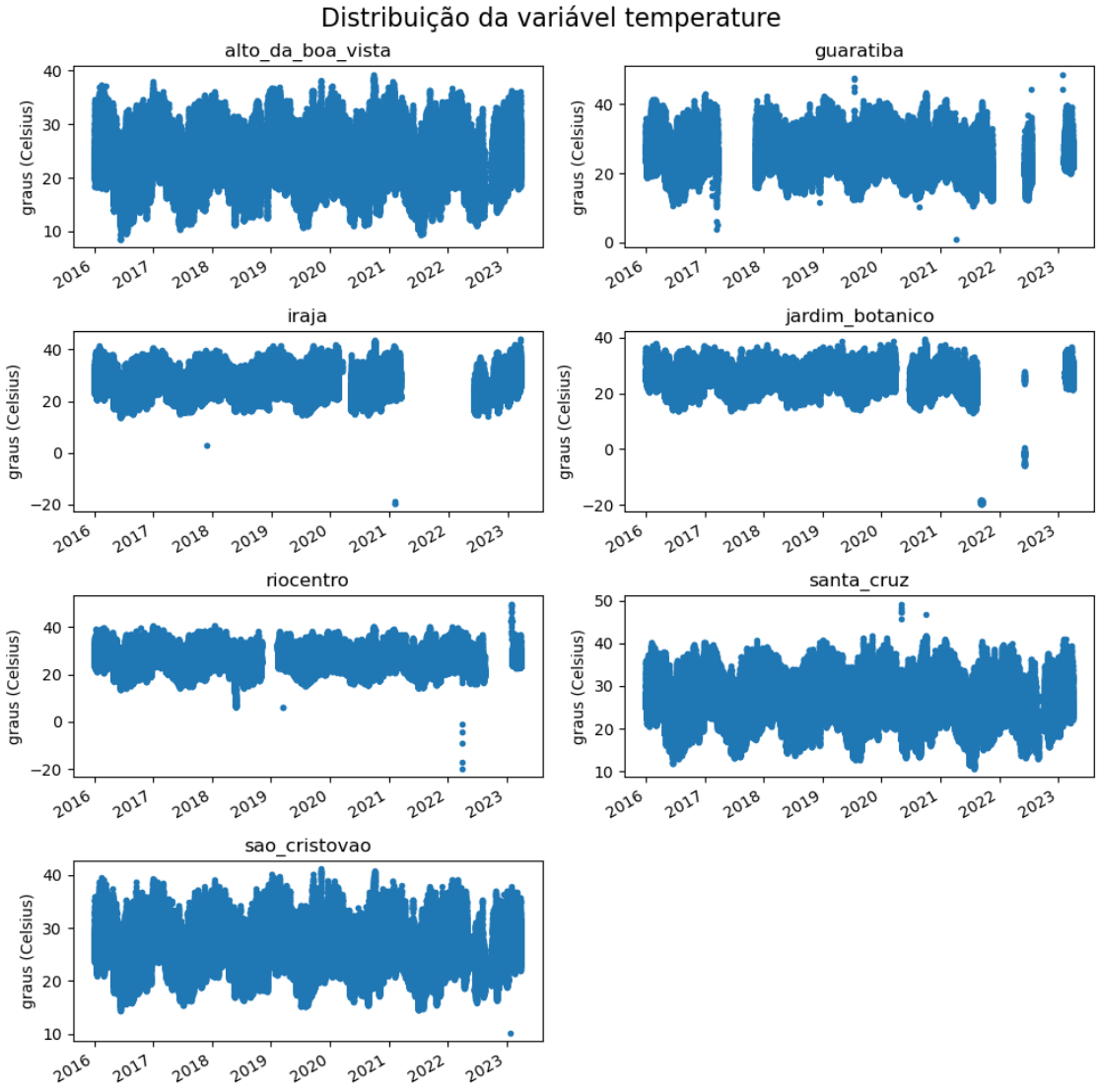


Figura V.8: Distribuição da variável *temperature* para as estações meteorológicas do Alerta Rio. Alguns valores estão fora dos limites padrões da distribuição histórica dos dados.

trabalho, uma vez que *precipitation* é a variável alvo.

Não foram encontrados dados espúrios. A Figura V.10 apresenta o histograma da variável *precipitation* considerando os dados de todas as estações. Na Figura V.10a é apresentado o histograma para todo o conjunto de dados. Na Figura V.10b os valores de precipitação 0.0 são removidos. Como é possível observar, a variável alvo *precipitation* é significativamente desbalanceada, o que torna o aprendizado por AM particularmente desafiador.

V.2 Construção do Grafo

Antes de prosseguir com a integração e o pré-processamento das fontes de dados, é necessário definir as coordenadas de uma grade regular $\mathbf{\Gamma}$ de pontos a serem interpolados. A decisão sobre $\mathbf{\Gamma}$ tem implicação direta sobre a integração e o pré-processamento dos dados, conforme é apresentado

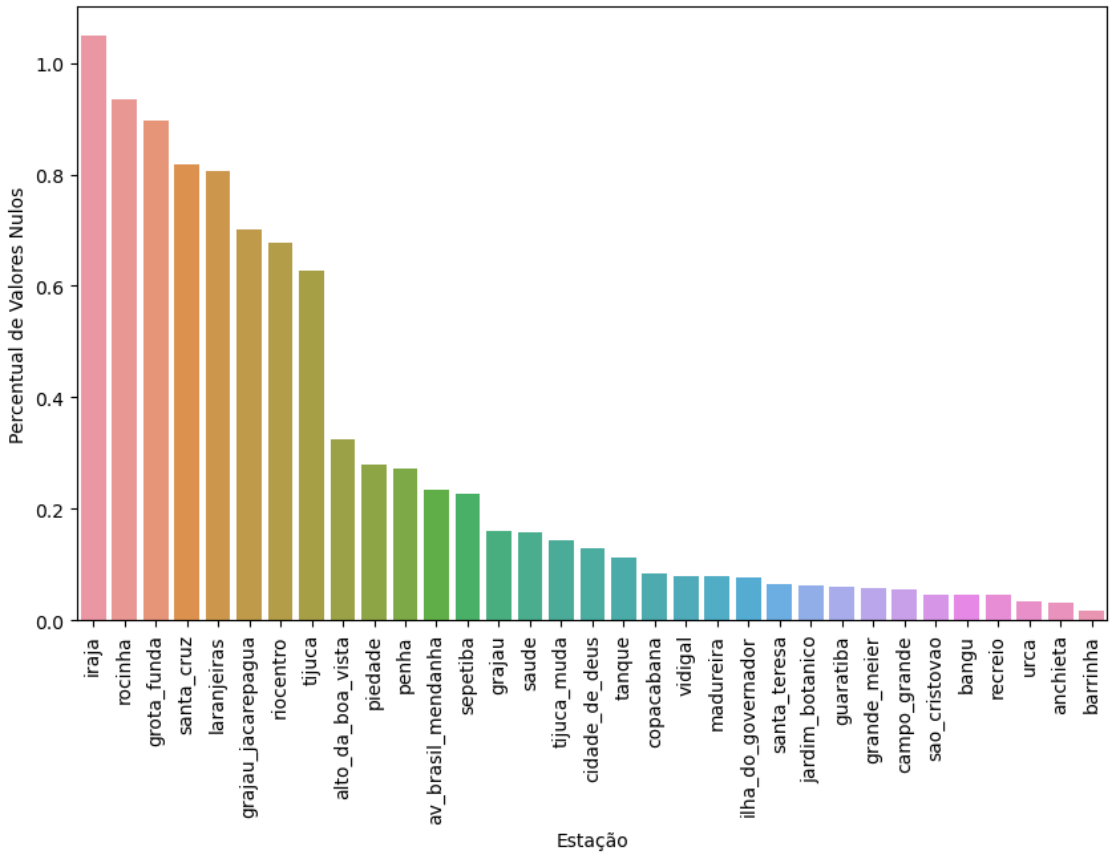
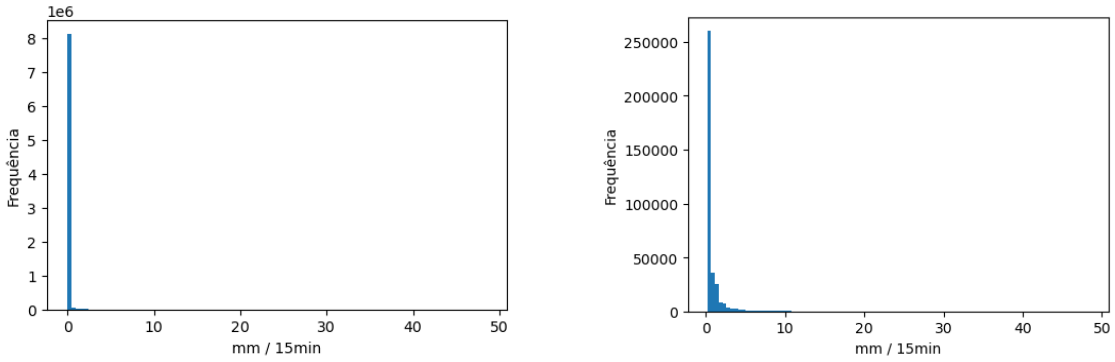


Figura V.9: Percentual de valores nulos da precipitação observada pelas estações pluviométricas do Alerta Rio.



(a) *precipitation* (b) *precipitation* > 0.0

Figura V.10: Os gráficos apresentam o histograma para a variável *precipitation* da fonte de dados Alerta Rio. Em (a) é apresentado o histograma para todo o conjunto de dados. Em (b) os valores de precipitação 0.0 são removidos. Os gráficos mostram o significativo desbalanceamento dessa variável.

na Seção V.3. A Figura V.11 apresenta os pontos da grade regular $\mathbf{\Gamma}$ arbitrariamente definida para o experimento. Os pontos da grade $\mathbf{\Gamma}$ possuem aproximadamente 3 km de distância em longitude e $3,3$ km de distância em latitude.

O conjunto de coordenadas $\mathbf{P}$ de estações telemétricas é adicionado à grade $\mathbf{\Gamma}$, gerando a grade irregular $\mathbf{\Upsilon} = \mathbf{\Gamma} \cup \mathbf{P}$. Cabe ressaltar que neste exemplo consideramos $\mathbf{P}$ o conjunto de todas

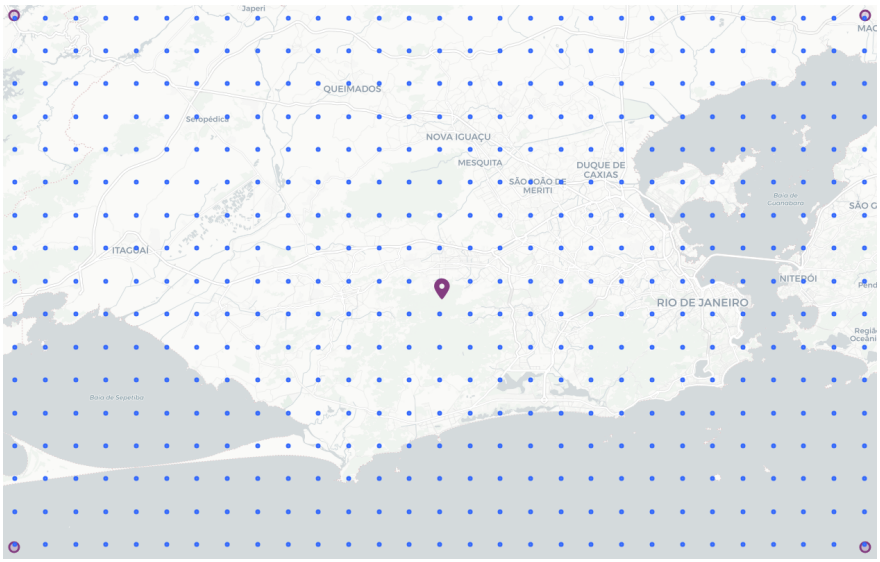


Figura V.11: Grade regular $\mathbf{\Gamma}$ de pontos. Os pontos em azul representam as coordenadas para as quais a STGCN deve inferir valores de precipitação interpolados. Os quatro pontos roxos nas extremidades representam os limites da área territorial de estudo deste trabalho.

as estações telemétricas empregadas neste trabalho. No entanto, a Seção V.3 explica como $\mathbf{P}$ é derivado em função das estações selecionadas para um determinado experimento de *cross-validation*.

O conjunto de coordenadas $v \in \mathbf{\Upsilon}$ representam os vértices do grafo. As arestas são definidas por um limiar de distância $q = 3,4$ km entre as coordenadas, aplicando a distância *Haversine*. O grafo resultante é apresentado na Figura V.12, em que é possível distinguir entre os pontos da grade $\mathbf{\Gamma}$ distribuídos uniformemente e os pontos que representam as estações telemétricas.

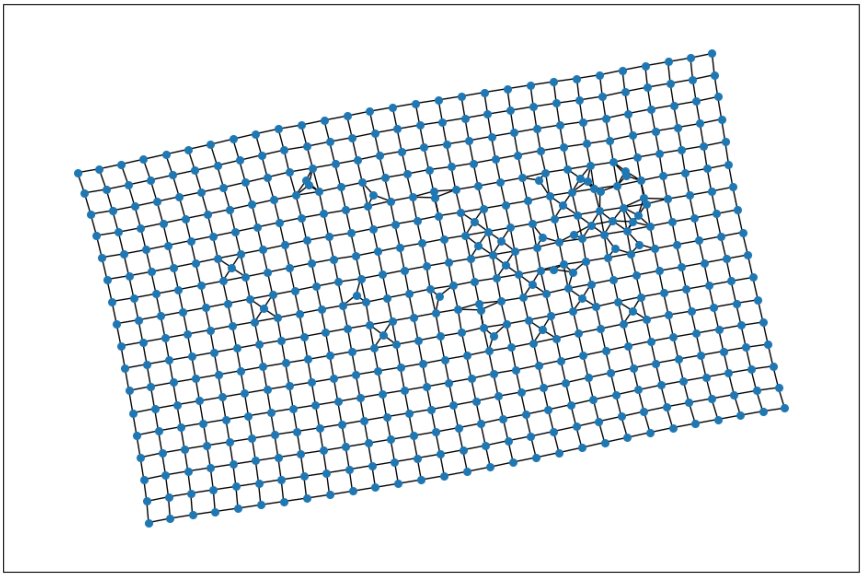


Figura V.12: Grafo construído com base na grade irregular $\mathbf{\Upsilon}$ de pontos. Cada vértice representa uma coordenada geográfica. As arestas foram definidas por um limiar de distância de $3,4$ km.

V.3 Pré-processamento dos Dados

Neste trabalho definimos três etapas de mais alto nível para o processamento dos dados. A primeira etapa consiste na *otimização* e *curadoria* dos dados. Esta primeira etapa é independente de integração, logo é processada separadamente para cada instituição/instrumento meteorológico. A segunda etapa consiste no *pré-processamento* e *integração*. Esta etapa processa transformações específicas deste trabalho e também transformações comuns em AM (*e.g.* imputação, normalização), além de integrar as fontes de dados. A terceira etapa consiste em aplicar a técnica *janelas deslizantes* para criar os exemplos de treino e teste na forma de séries temporais de grafos.

O processo para operar a etapa de *otimização* e *curadoria* dos dados e é detalhada na Seção IV.3. As transformações foram aplicadas de acordo com os problemas encontrados na análise exploratória detalhada na Seção V.1. A data/hora dos registros foram padronizados para o *timezone* UTC. A variável *wind_speed* foi convertida para a unidade m/s (metros por segundo). Dados espúrios foram removidos, *i.e.*, transformados para valores nulos. Dados espúrios são aqueles registrados com valores fora dos limites definidos na Tabela V.7, conforme consultoria prestada pelos meteorologistas do Alerta Rio.

Tabela V.7: Valores limites das variáveis meteorológicas.

Variável	Valor Mínimo	Valor Máximo
precipitation (acc 15min)	0.0	70.0
precipitation (acc 1h)	0.0	180.0
wind_speed (km/h)	0.0	120.0
wind_speed (m/s)	0.0	34.0
wind_dir (degrees)	0.0	360.0
temperature (Celsius)	0.0	50.0
humidity (%)	0.0	100.0
pressure (mB)	800.0	1200.0
dew_point (Celsius)	0.0	40.0

A etapa de *pré-processamento* e *integração* dos dados inicia com a definição de um conjunto $\mathcal{I}$ de *Estações de Interesse*. A coordenadas das estações do conjunto $\mathcal{I}$ simulam coordenadas sem informação e que se deseja interpolar. Neste trabalho selecionamos seis estações para compor $\mathcal{I}$. As estações selecionadas são as que apresentam a maior quantidade de exemplos de eventos mais extremos de precipitação e são listadas na Tabela V.8.

Para cada estação $\mathcal{I}^{(i)} \in \mathcal{I}$ é definido um conjunto $\mathcal{N}^{(i)}$ de *Estações Vizinhas* próximas. As estações $\mathcal{N}^{(i)}$ são as identificadas em um raio de 15 km em torno de $\mathcal{I}^{(i)}$. $\mathcal{N}^{(i)}$ é ordenado de forma crescente de distância da estação $\mathcal{I}^{(i)}$, *i.e.*, da mais próxima para a mais distante. O conjunto $\mathbf{P}_j^{(i)} = \{\mathcal{I}^{(i)} \cup \mathcal{N}_j^{(i)}\}$, sendo $j = \{1, 2, \cdots, |\mathcal{N}^{(i)}|\}$ é então composto por $\left[\{\mathcal{N}_1^{(i)}, \mathcal{N}_2^{(i)}, \cdots, \mathcal{N}_j^{(i)}\} \cup \mathcal{I}^{(i)}\right]$. Dessa forma, $\mathbf{P}_j^{(i)}$ define as coordenadas de estações que são adi-

Tabela V.8: Estações de Interesse. Simulam uma coordenada a ser interpolada.

Id	Estação
1	Tijuca Muda
2	Guaratiba
3	Alto da Boa Vista
4	Campo Grande
5	Grota Funda
6	Riocentro

cionadas à grade regular $\mathbf{\Gamma}$ para compor $\mathbf{\Upsilon}_j^{(i)} = \mathbf{\Gamma} \cup \mathbf{P}_j^{(i)}$. Neste trabalho, analisamos a influência da vizinhança de $\mathcal{I}^{(i)}$ treinando e avaliando modelos para cada $\mathbf{\Upsilon}_j^{(i)}$. Esta análise é detalhada na Seção V.4. As etapas de processamento de dados a seguir detalhadas são processadas para cada conjunto de coordenadas $\mathbf{\Upsilon}_j^{(i)}$.

V.3.1 Pré-processamento de Dados de Estações Telemétricas

Os dados das estações em $\mathbf{P}_j^{(i)}$ são filtrados de acordo com as datas $\mathcal{D}$ contendo dados de radar e detalhadas na Seção V.1. A resolução temporal das estações do INMET e do Alerta Rio são, respectivamente, 1 hora e 15 minutos. Por conta disso, os dados das estações do Alerta Rio foram agregados para 1 hora, sendo a variável *precipitation* agregada pela soma e as demais variáveis meteorológicas pela média. Em seguida são removidos os registros em que a variável *precipitation* é nula. Decidimos não realizar imputação sobre a variável alvo. Para cada data/hora t verificamos se existem registros associados para todas as coordenadas $\mathbf{P}_j^{(i)}$. Em caso negativo, excluímos todos os registros associados a t. Isso se justifica pelo fato que empregamos grafos estáticos neste trabalho, logo a topologia do grafo precisa ser idêntica em todo o conjunto de dados.

Os dados de $\mathcal{I}^{(i)}$ são removidos do conjunto de dados, uma vez que simula a coordenada sem informação, evitando o vazamento de dados. Na prática, apenas as variáveis *features* são removidas e a variável alvo *precipitation* é mascarada, viabilizando a etapa final de processamento que aplica a técnica *janelas deslizantes*. O conjunto é divido em dados de treino e teste antes de processar outras transformações para também evitar o vazamento de dados. Registros anteriores à 01/02/2022 foram separados para o treino e igual ou posterior para o teste.

A variável *wind_dir* é tratada, *i.e.*, aplicamos o valor 0.0 quando *wind_speed* é 0.0. As variáveis *wind_speed* e *wind_dir* são transformadas para um padrão cíclico[3]. A informação de data/hora da observação é transformada em *feature* pelo método de transformação usando o seno e cosseno[4]. Todas as *features*, com exceção da variável alvo *precipitation*, são normalizadas pela função *Min-max*[5]. As demais variáveis têm os valores ausentes imputados. Para a imputação empregamos

[3]https://unidata.github.io/MetPy/latest/api/generated/metpy.calc.wind_components.html
[4]https://www.kaggle.com/code/avanwyk/encoding-cyclical-features-for-deep-learning
[5]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

o *KNNImputer*[6] da biblioteca *scikit-learn* que emprega o algoritmo *k-Nearest Neighbors* neste processo.

V.3.2 Pré-processamento de Dados de Radar

Conforme citado na Seção V.1.1 empregamos a variável *reflectivity* do radar que é agregada pela média para uma resolução espacial de 1 km (latitude, longitude e altitude) e temporal de 1 hora, já na etapa de *otimização*. Essa decisão se apoia em duas justificativas. A primeira é viabilizar a análise exploratória dos dados em um conjunto de alta resolução espacial e temporal. A segunda é que os dados das estações telemétricas do INMET são disponibilizados em resolução temporal horária.

Os dados de radar são filtrados para conter somente observações dentro dos limites de coordenadas da grade regular $\mathbf{\Gamma}$. Devido ao efeito do *cone do silêncio*, os dados são filtrados para conter somente os registros observados até 2 km de altitude. Em seguida é processada a agregação que emprega a grade irregular $\mathbf{\Upsilon}_j^{(i)}$. Para cada passo de tempo t, cada coordenada $v \in \mathbf{\Upsilon}_j^{(i)}$ é agregada pelos valores de *reflectivity* em um raio de $1,5$ km. No processo de agregação são geradas as *features média*, *mínimo*, *máximo*, *primeiro quartil*, *mediana* e *terceiro quartil* em relação à refletividade observada dentro do raio.

Após a agregação a informação de data/hora da observação é transformada em *feature* pelo método de transformação usando o seno e cosseno. Em seguida as *features* são normalizadas pela função *Min-max*. Não foi necessário aplicar técnicas de imputação já que não haviam dados faltantes, o que é esperado devido à alta resolução dos dados brutos.

V.3.3 Integração dos Dados e Construção das Séries Temporais de Grafos

A integração dos dados pré-processados é processada aplicando *outer join* por (coordenada, data/hora). A integração gera dados faltantes uma vez que a maioria das coordenadas não possuem valores observados para todas as variáveis meteorológicas. Para esses casos, imputamos o valor 0.0 e incluímos uma *flag* binária marcadora de preenchimento que distingue os valores 0.0 verdadeiros.

Para ilustrar o processo de integração, é apresentado um exemplo trivial. Considere as Tabelas V.9 e V.10 que representam, respectivamente, os dados de radar e os dados de estações telemétricas. Cabe ressaltar que este é um exemplo e que as colunas de dados não refletem os dados empregados neste trabalho. A coluna *location_id* representa uma coordenada geográfica. As colunas r, w e p representam, respectivamente, dados de *reflectivity*, *wind_speed* e *precipitation*. Como é possível observar, não existe uma estação telemétrica na coordenada *location_id* $= 2$.

Ao aplicar o *outer join*, temos como resultado a Tabela V.11. Essa integração gera valores

[6] https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html

Tabela V.9: Exemplo de Integração de Dados - Tabela A.

datetime	location_id	r
2023-10-01 12:00:00	1	35.50
2023-10-01 12:00:00	2	40.00

Tabela V.10: Exemplo de Integração de Dados - Tabela B.

datetime	location_id	w	p
2023-10-01 12:00:00	1	0.00	5.00

ausentes para a coordenada *location_id* = 2. Para contornar o problema, imputamos valores 0.0 e adicionamos a *flag* binária marcadora de preenchimento que distingue os valores 0.0 verdadeiros. O resultado é apresentado na Tabela V.12.

Tabela V.11: Exemplo de Integração de Dados - Tabela C.

datetime	location_id	r	w	p
2023-10-01 12:00:00	1	35.50	0.00	5.00
2023-10-01 12:00:00	2	40.00	NaN	NaN

Tabela V.12: Exemplo de Integração de Dados - Tabela D.

datetime	location_id	r	w	w_flag	p	p_flag
2023-10-01 12:00:00	1	35.50	0.00	True	5.00	True
2023-10-01 12:00:00	2	40.00	0.00	False	0.00	False

O conjunto de dados integrado consiste de séries temporais multivariadas para múltiplas coordenadas geográficas. Cada coordenada está associada à um vértice do grafo $\mathcal{G}_j^{(i)}$ construído a partir de $\boldsymbol{\Upsilon}_j^{(i)}$. Conforme definido neste trabalho, $\mathcal{G}_j^{(i)}$ é um grafo estático. Logo, para cada passo de tempo t temos o grafo $\mathcal{G}_j^{(i)} = (\mathbf{X}_t, \mathbf{A}^{(\mathbf{i},\mathbf{j})})$ em que $\mathbf{A}^{(\mathbf{i},\mathbf{j})}$ é matriz de adjacência que representa $\mathcal{G}_j^{(i)}$ e $\mathbf{X}_t \in \mathbb{R}^{N \times F}$ é a matriz de *features* no tempo t.

Transformar o conjunto de dados em séries temporais de grafos consiste em aplicar a técnica *janelas deslizantes* detalhada na Seção II.7. Em nosso trabalho empregamos o tamanho de janela igual a 6, *i.e.*, a técnica produz M séries temporais, cada $m \in M$ contendo dados meteorológicos das últimas 6 horas. Dessa forma, cada série $m \in M$ produzida pela técnica é um exemplo de entrada para o modelo e pode ser denotado por $\{\mathbf{X}_{(t-5)}, \mathbf{X}_{(t-4)}, \ldots, \mathbf{X}_{(t)}\}$. A quantidade de exemplos de treinamento e teste varia em função de cada definição de $\boldsymbol{\Upsilon}$.

V.4 Treinamento e Avaliação do Modelo

O objetivo deste trabalho é avaliar o aprendizado da interpolação de precipitação por redes neurais em grafos. Na análise dos resultados, comparamos o valor real observado por uma estação $\mathcal{I}^{(i)}$ e o predito pelo modelo. Também comparamos o nosso resultado com o método estatístico IDW. Como objetivos secundários analisamos duas questões: *i)* o impacto sobre a acurácia quando empregada a *feature reflectivity* e; *ii)* a influência dos dados de *Estações Vizinhas* (EV) em uma determinada coordenada de interesse.

Para cada estação $\mathcal{I}^{(i)}$ definida na Tabela V.8 processamos $K = |\mathcal{N}^{(i)}|$ conjuntos de dados distintos, conforme detalhado na Seção V.3. Então temos que $\mathcal{P}_1^{(i)} = \{\mathcal{I}^{(i)}, \mathcal{N}_1^{(i)}\}$, $\mathcal{P}_2^{(i)} = \{\mathcal{I}^{(i)}, \mathcal{N}_1^{(i)}, \mathcal{N}_2^{(i)}\}$ e $\mathcal{P}_K^{(i)} = \{\mathcal{I}^{(i)}, \mathcal{N}_1^{(i)}, \mathcal{N}_2^{(i)}, \ldots, \mathcal{N}_K^{(i)}\}$. Vale lembrar que o conjunto $\mathcal{N}^{(i)}$ é ordenado de forma crescente de distância da estação $\mathcal{I}^{(i)}$, *i.e.*, da mais próxima para a mais distante. Dessa forma, podemos afirmar que os dados de um novo vizinho são adicionados a cada nova versão de conjunto de dados, cumulativamente aos já adicionados no conjunto anterior. Tal configuração permite treinar modelos para cada versão do conjunto de dados, viabilizando a análise de influência das EV.

Ainda, para cada versão de conjunto de dados treinamos dois modelos STGCN. Um em que empregamos todas as *features* disponíveis em $\boldsymbol{\Upsilon}_j^{(i)}$ e outro em que removemos a *feature reflectivity*. Dessa forma, analisamos o impacto dessa *feature* sobre a acurácia do modelo.

A STGCN selecionada para treinar o modelo é detalhada na Seção IV.4. Empregamos o *framework PyTorch Geometric Temporal*[7] para implementar a rede neural com três camadas. A primeira camada implementa a STGCN que aprende as relações espaço-temporais, gerando o vetor de contexto da série temporal de entrada. As duas camadas finais implementam MLP para processar as correlações. A função $ReLU$ é utilizada como função de ativação das camadas internas. Como função de custo é empregada a *Mean Squared Error*. A taxa de aprendizado é configurada para 1×10^{-3} e a rede é treinada por 1000 épocas com *early stopping* de 100 épocas e *batches* de tamanho 32.

[7] https://pytorch-geometric-temporal.readthedocs.io/en/latest/modules/root.html

Capítulo VI Resultados e Análise

Este Capítulo está organizado como segue. A Seção VI.1 inicia com uma visão geral. As Seções VI.1.1, VI.1.2 e VI.1.3 apresentam, respectivamente, os resultados obtidos para as EIs *Tijuca Muda*, *Guaratiba* e *Alto da Boa Vista*. A Seção VI.1.4 resume os resultados obtidos para as demais EIs. Por fim, a Seção VI.2 apresenta uma análise retrospectiva sobre os resultados obtidos.

VI.1 Resultados

Esta seção apresenta os resultados obtidos pelos experimentos detalhados no Capítulo V. Comparamos a acurácia alcançada pelo modelo multivariado treinado com STGCN com a do método estatístico IDW. Também apresentamos a variação da acurácia do modelo STGCN sob dois aspectos: *i)* em função das informações da vizinhança e; *ii)* em função do emprego da *feature reflectivity*.

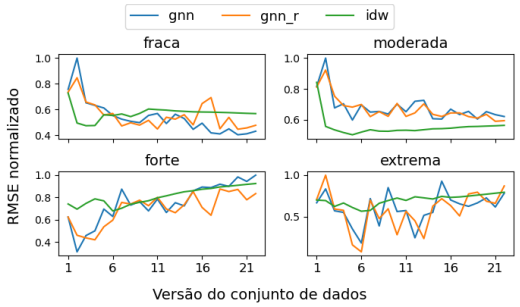
Esta seção está dividida em quatro subseções. As Seções VI.1.1, VI.1.2, VI.1.3 apresentam, respectivamente, os resultados para as EI *Tijuca Muda*, *Guaratiba* e *Alto da Boa Vista*, estações para as quais o modelo STGCN obteve melhores resultados. A Seção VI.1.4 apresenta um consolidado das demais EI listadas na Tabela V.8. A análise retrospectiva acerca dos resultados é apresentada na Seção VI.2. Quando adequado, empregamos a coluna *Id* da Tabela V.8 para denotar o índice i nos conjuntos $\mathcal{I}$ e $\mathcal{N}$. Por exemplo, $\mathcal{I}^{(1)}$ representa a estação *Tijuca Muda* e $\mathcal{N}^{(1)}$ o conjunto de EVs de $\mathcal{I}^{(1)}$.

Este trabalho considerou seis EI e para cada uma delas foram processados $|\mathcal{N}^{(i)}|$ conjuntos de dados distintos. Para cada conjunto de dados foram treinados dois modelos STGCN, um empregando a *feature reflectivity* e o outro não. Logo, para cada EI temos $|\mathcal{N}^{(i)}| \times 2$ modelos treinados, sendo que $|\mathcal{N}^{(i)}|$ varia em função de $\mathcal{I}^{(i)}$. No total foram treinados 140 modelos, todos com os hiperparâmetros fixados conforme descrito na Seção V.4.

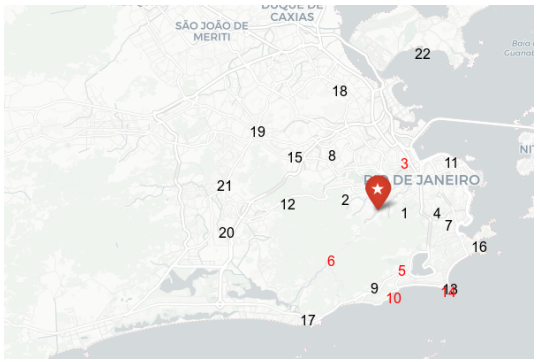
Por conta da variedade de modelos STGCN treinados, apresentamos somente os resultados para o melhor modelo em termos de acurácia. Escolher o melhor modelo é uma questão subjetiva, já que analisamos os resultados por classes de níveis de precipitação apresentados na Tabela V.1. Sendo assim, um modelo pode ter boa acurácia em um determinado nível de precipitação $\mathbf{C}_i$ e não ter em um nível $\mathbf{C}_j$. Quando isso acontece, apresentamos o modelo que tem melhor equilíbrio de acurácia entre as classes.

VI.1.1 Estação Tijuca Muda

A Figura VI.1 apresenta a localização das EVs de $\mathcal{I}^{(1)}$ e o gráfico de variação na acurácia dos modelos a medida que adicionamos dados de vizinhos cada vez mais distantes. A Figura VI.1a apresenta a variação na acurácia para os modelos STGCN e para o método IDW. O eixo x representa o índice j em $\mathcal{N}_j^{(1)}$, *i.e.*, a versão do conjunto de dados. O modelo *gnn_r* é treinado com a *feature reflectivity* e o *gnn* sem. A Figura VI.1b apresenta a localização das estações na cidade do Rio de Janeiro. O ícone de localização com estrela representa a EI. Os números representam as EVs em ordem crescente de distância da EI. Números vermelhos denotam uma *Estação Meteorológica* e números pretos uma *Estação Pluviométrica*. A relação das estações $\mathcal{N}^{(1)}$ é apresentada no Apêndice A.1.



(a) influência da vizinhança



(b) localização das estações

Figura VI.1: EI *Tijuca Muda* (a) O gráfico apresenta a variação na acurácia do modelo. O eixo x representa o índice j em $\mathcal{N}_j^{(i)}$, *i.e.*, a versão do conjunto de dados. O modelo *gnn_r* é treinado com a *feature reflectivity* e o *gnn* sem. (b) Localização das estações. O ícone de localização com estrela representa a EI. Os números representam as EVs em ordem crescente de distância da EI. Números vermelhos denotam uma *Estação Meteorológica* e números pretos uma *Estação Pluviométrica*.

Em relação aos modelos STGCN, nas categorias *fraca* e *moderada* o erro tende a diminuir com a adição de EVs. Na categoria *extrema*, o erro diminui significativamente até a adição da sexta EV e tende a aumentar após. Na categoria *forte* ocorre o mesmo até a segunda EV. Já para o método IDW, o erro tende a permanecer constante conforme a adição de EVs mais distantes, o que é esperado dada a formulação matemática do próprio método, que pondera inversamente as informações de EVs mais distantes. A Figura VI.2 apresenta as curvas de aprendizado para o modelo *gnn_r* que mostram boa convergência e generalização do modelo.

A Tabela VI.1 apresenta o *RMSE* por modelo e categoria para cada $\mathcal{N}^{(i)}$. Os índices das colunas representam o índice j em $\mathcal{N}_j^{(1)}$. Por restrição de espaço, apresentamos somente os valores para as dez primeiras EVs.

Novamente observamos que os modelos *gnn* e *gnn_r* apresentam resultados melhores que o IDW em eventos de precipitação *forte* e *extrema*, com poucas exceções. Em alguns casos o erro é

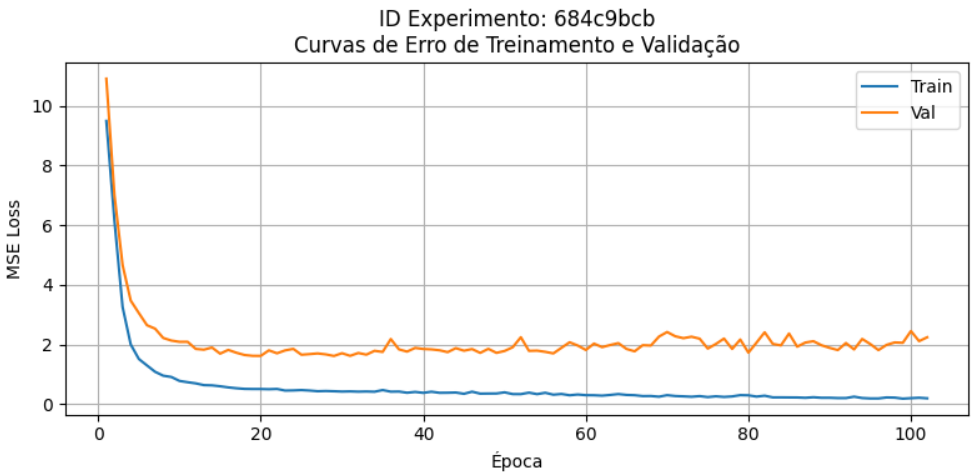


Figura VI.2: Curvas de Aprendizado - Tijuca Muda

Tabela VI.1: Tijuca Muda - RMSE.

Categoria	Modelo	1	2	3	4	5	6	7	8	9	10
fraca	gnn	0.831	1.098	0.716	0.694	0.672	**0.608**	0.577	0.558	0.547	0.608
	gnn_r	0.808	0.930	0.723	0.699	**0.612**	0.625	**0.519**	**0.546**	**0.526**	**0.567**
	idw	**0.799**	**0.544**	**0.521**	**0.522**	0.614	**0.608**	0.621	0.599	0.626	0.663
moderada	gnn	4.831	5.922	4.007	4.167	3.541	4.122	3.846	3.871	3.771	4.150
	gnn_r	**4.811**	5.463	4.457	4.097	4.037	4.140	3.673	3.864	3.684	4.179
	idw	4.989	**3.296**	**3.164**	**3.061**	**2.975**	**3.077**	**3.169**	**3.115**	**3.110**	**3.144**
forte	gnn	**9.508**	**4.740**	6.977	7.621	10.608	9.590	13.340	**11.149**	11.633	**10.354**
	gnn_r	9.521	7.015	**6.655**	**6.395**	**8.177**	**9.090**	11.519	11.298	11.825	11.065
	idw	11.307	10.598	11.385	12.006	11.732	10.351	**10.725**	11.274	**11.533**	11.748
extrema	gnn	**17.558**	21.919	**15.017**	**14.520**	9.305	4.857	18.869	**10.246**	22.340	14.787
	gnn_r	18.627	26.258	15.587	15.179	**4.269**	**2.073**	18.373	12.568	**15.652**	**7.467**
	idw	18.400	**18.241**	16.403	17.477	16.069	14.881	**15.166**	17.400	18.250	19.052

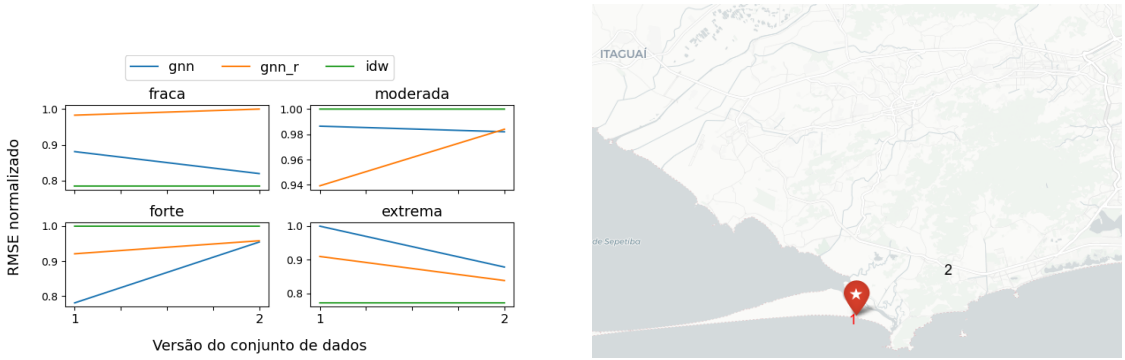
significativamente menor, chegando a um erro aproximadamente 86% menor que o IDW para chuva *extrema* no conjunto 6. Na categoria *moderada* o método IDW obtém melhores resultados, exceto no conjunto 1. Na categoria *fraca* os modelos STGCN passam a ter resultados igual ou melhor que o IDW a partir do conjunto 6. Apesar disso, a diferença de erro entre os métodos nas categorias *fraca* e *moderada* é relativamente pequena em quase todos os conjuntos.

Quanto ao emprego da *feature reflectivity*, a Tabela VI.1 apresenta que o modelo treinado com a inclusão dessa *feature* retorna resultados melhores em alguns casos. Na categoria *extrema* os conjuntos 5, 6, 9 e 10 apresentam erro significativamente menor quando empregada a *feature reflectivity*. Já no conjunto 2 ocorre o inverso. Nas demais categorias a diferença de erro entre os modelos *gnn* e *gnn_r* é, em geral, pequena.

VI.1.2 Estação Guaratiba

A Figura VI.3 apresenta a localização das EVs de $\mathcal{I}^{(2)}$ e o gráfico de variação na acurácia dos modelos. Em particular, a estação $\mathcal{I}^{(2)}$ possui apenas dois vizinhos no raio de 15 km a partir de

sua localização. Também cabe destacar que seu vizinho mais próximo dista apenas 80 metros de sua posição. A relação das estações $\mathcal{N}^{(2)}$ é apresentada no Apêndice A.2.



(a) influência da vizinhança (b) localização das estações

Figura VI.3: EI *Guaratiba* (a) O gráfico apresenta a variação na acurácia do modelo. O eixo x representa o índice j em $\mathcal{N}_j^{(i)}$, *i.e.*, a versão do conjunto de dados. O modelo *gnn_r* é treinado com a *feature reflectivity* e o *gnn* sem. (b) Localização das estações. O ícone de localização com estrela representa a EI. Os números representam as EVs em ordem crescente de distância da EI. Números vermelhos denotam uma *Estação Meteorológica* e números pretos uma *Estação Pluviométrica*.

Na Figura VI.3a observamos que a adição do segundo vizinho impacta negativamente a acurácia dos modelos STGCN nas categorias *moderada* e *forte*. O inverso ocorre na categoria *extrema*. O método IDW é praticamente constante, algo esperado já que o segundo vizinho adicionado fica à aproximadamente $8,49$ km e, conforme a formulação do método, recebe um peso significativamente menor em relação ao vizinho mais próximo à 80 metros. A Figura VI.4 apresenta as curvas de aprendizado para o modelo *gnn_r* que mostram boa convergência e generalização do modelo.

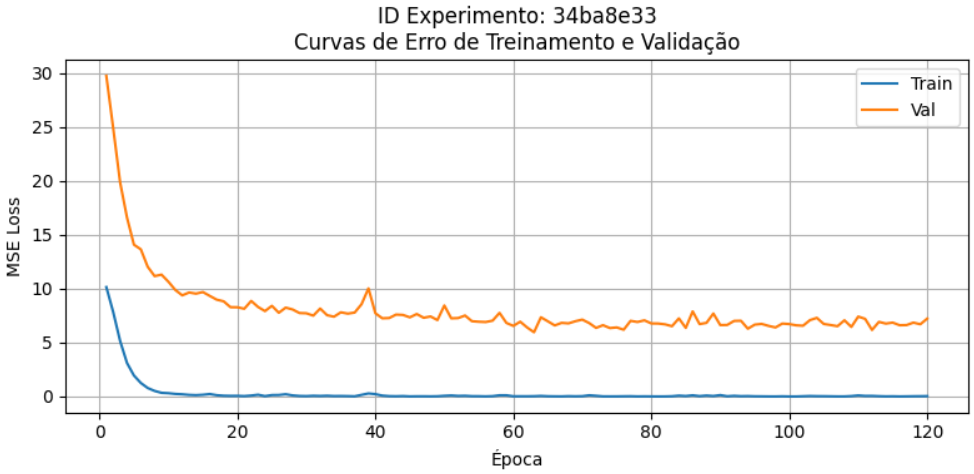


Figura VI.4: Curvas de Aprendizado - Guaratiba

Na Tabela VI.2 observamos que os modelos STGCN apresentam melhor resultado nas categorias *moderada* e *forte*, quanto o IDW apresenta erro menor nas categorias *fraca* e *extrema*. Apesar disso, a diferença de erro entre os métodos é, em geral, pequena. O emprego da *feature reflectivity*

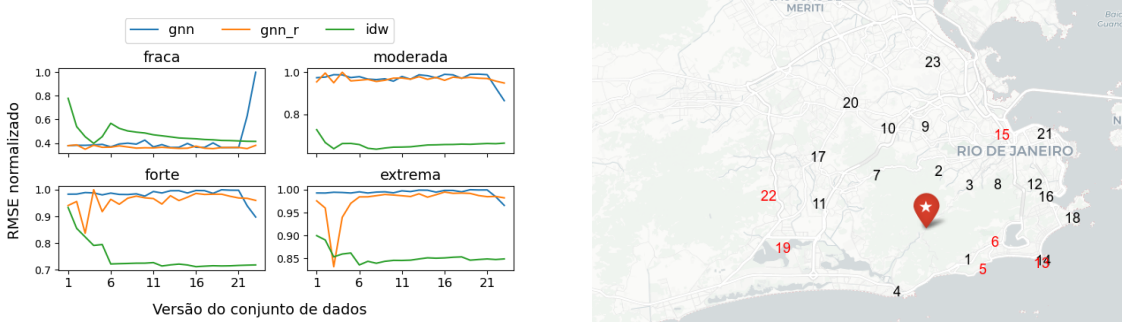
aumentou a acurácia nas categorias *moderada* e *extrema* no caso do conjunto de dados 1.

Tabela VI.2: Guaratiba - RMSE.

		1	2
Categoria	Modelo		
fraca	gnn	0.181	0.168
	gnn_r	0.202	0.205
	idw	**0.161**	**0.161**
moderada	gnn	5.076	**5.054**
	gnn_r	**4.833**	5.064
	idw	5.146	5.146
forte	gnn	**16.089**	**19.673**
	gnn_r	18.982	19.748
	idw	20.608	20.607
extrema	gnn	39.983	35.124
	gnn_r	36.382	33.524
	idw	**30.868**	**30.867**

VI.1.3 Estação Alto da Boa Vista

A Figura VI.5 apresenta a localização das EVs de $\mathcal{I}^{(3)}$ e o gráfico de variação na acurácia dos modelos. A estação $\mathcal{I}^{(3)}$ é a que possui maior quantidade de vizinhos no raio considerado neste trabalho. A relação das estações $\mathcal{N}^{(3)}$ é apresentada no Apêndice A.3.



(a) influência da vizinhança

(b) localização das estações

Figura VI.5: EI *Alto da Boa Vista* (a) O gráfico apresenta a variação na acurácia do modelo. O eixo x representa o índice j em $\mathcal{N}_j^{(i)}$, *i.e.*, a versão do conjunto de dados. O modelo *gnn_r* é treinado com a *feature reflectivity* e o *gnn* sem. (b) Localização das estações. O ícone de localização com estrela representa a EI. Os números representam as EVs em ordem crescente de distância da EI. Números vermelhos denotam uma *Estação Meteorológica* e números pretos uma *Estação Pluviométrica*.

Na Figura VI.3a observamos que a adição de vizinhos quase não causa variação na acurácia dos modelos STGCN para as categorias *fraca* e *moderada*. Nas categorias *forte* e *extrema* os modelos *gnn_r* apresentam melhora na acurácia com a adição do vizinho 3. Já o método IDW tende a estabilizar a partir do vizinho 6. A Figura VI.4 apresenta as curvas de aprendizado para o modelo

gnn_r que mostram um aparente *overfitting*.

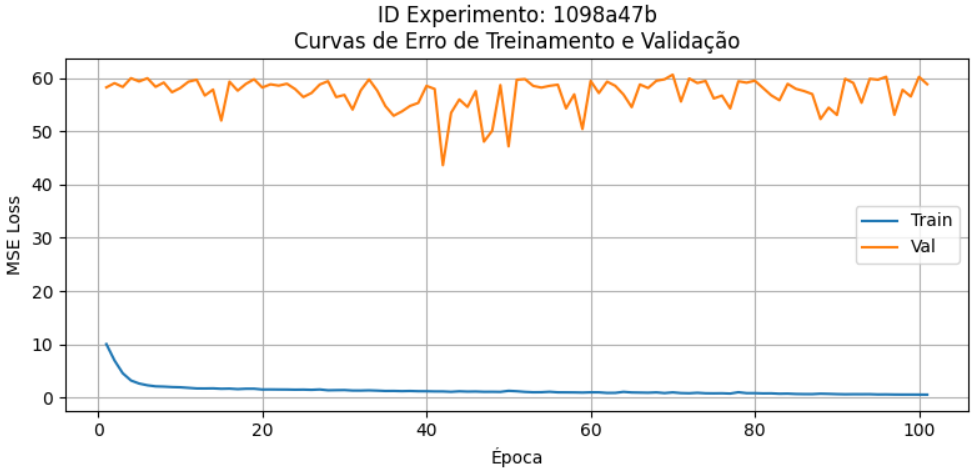


Figura VI.6: Curvas de Aprendizado - Guaratiba

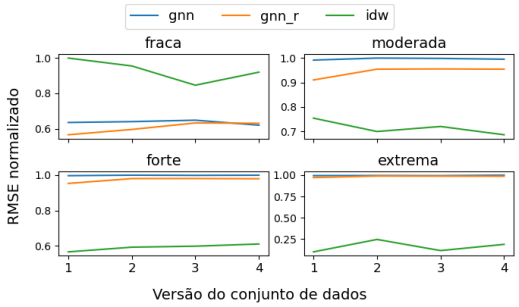
A Tabela VI.3 mostra que o método IDW apresenta erro menor em quase todos os casos das categorias *moderada*, *forte* e *extrema*. A exceção fica para um único caso no conjunto 3 para o modelo *gnn_r*, conforme também é observado na Figura VI.5a. Já na categoria *fraca* os modelos STGCN apresentam menor erro em todos os casos. O emprego da *feature reflectivity* no conjunto 3 proporciona um resultado aproximadamente 16% melhor que o modelo *gnn*. Para todos os demais casos há pouca diferença em empregar ou não essa *feature*.

Tabela VI.3: Alto da Boa Vista - RMSE.

Categoria	Modelo	1	2	3	4	5	6	7	8	9	10
fraca	gnn	**0.801**	**0.809**	0.809	0.815	0.826	0.781	0.833	0.847	0.830	0.902
	gnn_r	0.802	0.817	**0.738**	**0.810**	**0.773**	**0.775**	**0.799**	**0.779**	**0.758**	**0.764**
	idw	1.654	1.145	0.964	0.843	0.964	1.205	1.113	1.067	1.046	1.030
moderada	gnn	11.676	11.714	11.849	11.838	11.685	11.738	11.599	11.568	11.615	11.478
	gnn_r	11.440	11.944	11.386	11.988	11.494	11.532	11.578	11.462	11.532	11.652
	idw	**8.705**	**7.971**	**7.614**	**7.913**	**7.920**	**7.854**	**7.643**	**7.588**	**7.656**	**7.705**
forte	gnn	29.713	29.721	29.896	29.854	29.635	29.834	29.686	29.674	29.752	29.496
	gnn_r	28.431	28.887	25.291	30.216	27.734	29.130	28.573	29.261	29.491	29.302
	idw	**28.203**	**25.844**	**24.872**	**23.905**	**24.015**	**21.812**	**21.845**	**21.867**	**21.897**	**21.900**
extrema	gnn	83.465	83.452	83.600	83.554	83.456	83.674	83.456	83.628	83.677	83.492
	gnn_r	82.013	80.704	**69.922**	79.002	81.565	82.747	82.758	82.969	83.192	83.088
	idw	**75.624**	**74.832**	71.728	**72.249**	**72.436**	**70.243**	**70.905**	**70.552**	**70.909**	**71.080**

VI.1.4 Demais Estações

A Figuras VI.7, VI.8 e VI.9 apresentam a localização das EVs das estações $\mathcal{I}^{(4)}$, $\mathcal{I}^{(5)}$ e $\mathcal{I}^{(6)}$ e também os gráficos de variação na acurácia dos modelos. Em particular, as estações $\mathcal{I}^{(4)}$ e $\mathcal{I}^{(5)}$ apresentam baixa densidade espacial, além de quantidade reduzida de EVs em relação às outras EIs. As respectivas relações de EVs são apresentadas nos Apêndices A.4, A.5 e A.6.

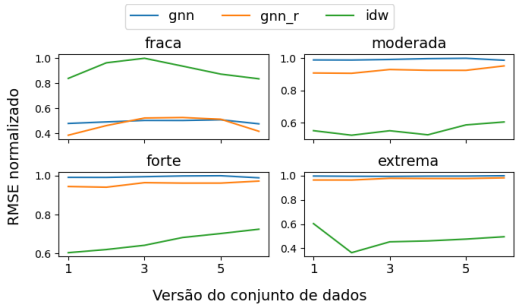


(a) influência da vizinhança

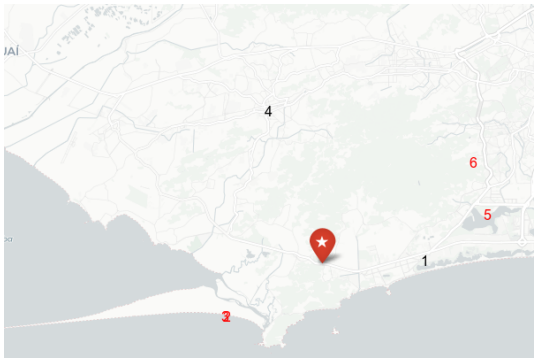


(b) localização das estações

Figura VI.7: EI *Campo Grande* (a) O gráfico apresenta a variação na acurácia do modelo. O eixo x representa o índice j em $\mathcal{N}_j^{(i)}$, *i.e.*, a versão do conjunto de dados. O modelo *gnn_r* é treinado com a *feature reflectivity* e o *gnn* sem. (b) Localização das estações. O ícone de localização com estrela representa a EI. Os números representam as EVs em ordem crescente de distância da EI. Números vermelhos denotam uma *Estação Meteorológica* e números pretos uma *Estação Pluviométrica*.



(a) influência da vizinhança

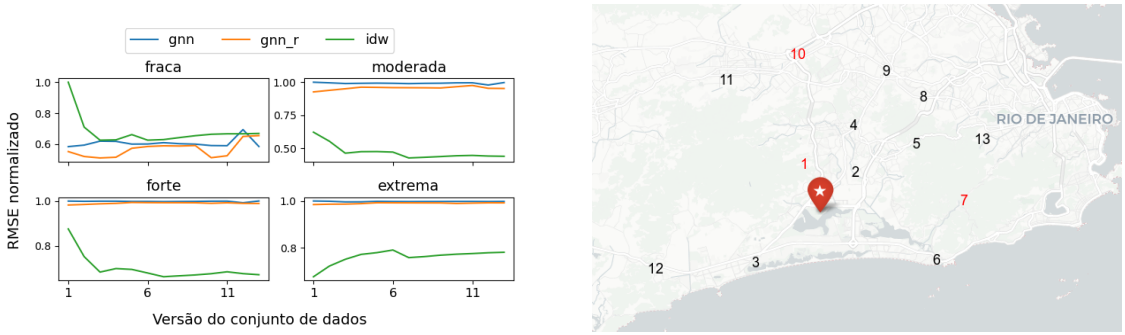


(b) localização das estações

Figura VI.8: EI *Grota Funda* (a) O gráfico apresenta a variação na acurácia do modelo. O eixo x representa o índice j em $\mathcal{N}_j^{(i)}$, *i.e.*, a versão do conjunto de dados. O modelo *gnn_r* é treinado com a *feature reflectivity* e o *gnn* sem. (b) Localização das estações. O ícone de localização com estrela representa a EI. Os números representam as EVs em ordem crescente de distância da EI. Números vermelhos denotam uma *Estação Meteorológica* e números pretos uma *Estação Pluviométrica*.

A análise sobre a influência da vizinhança para essas EIs é muito similar entre elas. Em todos os casos os modelos STGCN apresentam erro praticamente constante nas categorias *moderada*, *forte* e *extrema*. Ocorre praticamente o mesmo na categoria *fraca* para as EIs $\mathcal{I}^{(4)}$ e $\mathcal{I}^{(5)}$, enquanto que a $\mathcal{I}^{(6)}$ apresenta uma leve variação. As respectivas curvas de aprendizado dos modelos STGCN são apresentadas nas Figuras VI.10, VI.11 e VI.12 em que também observamos aparente *overfitting*.

Deixamos de discursar sobre o *RMSE* nesta seção, uma vez que os modelos não convergiram. Pelo mesmo motivo, deixamos de analisar o emprego da *feature reflectivity*. A Seção VI.2 aborda as prováveis causas desse problema.



(a) influência da vizinhança (b) localização das estações

Figura VI.9: EI *Riocentro* (a) O gráfico apresenta a variação na acurácia do modelo. O eixo x representa o índice j em $\mathcal{N}_j^{(i)}$, *i.e.*, a versão do conjunto de dados. O modelo *gnn_r* é treinado com a *feature reflectivity* e o *gnn* sem. (b) Localização das estações. O ícone de localização com estrela representa a EI. Os números representam as EVs em ordem crescente de distância da EI. Números vermelhos denotam uma *Estação Meteorológica* e números pretos uma *Estação Pluviométrica*.

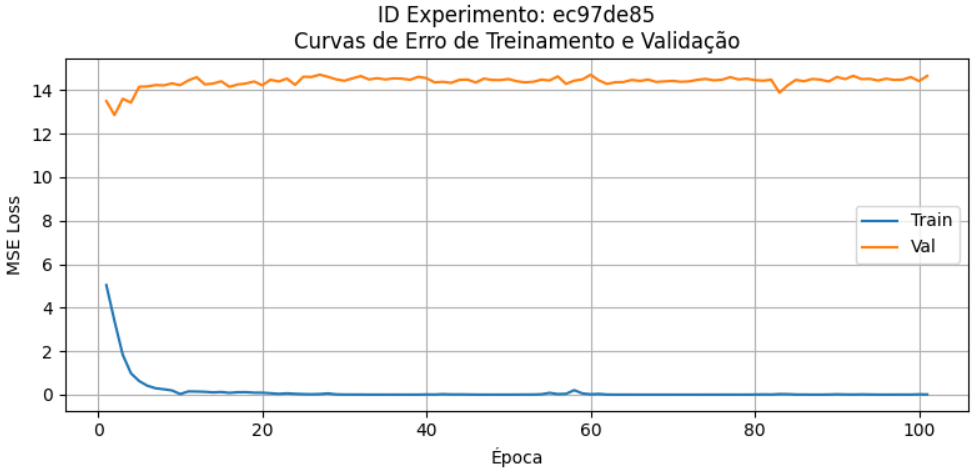


Figura VI.10: Curvas de Aprendizado - Campo Grande

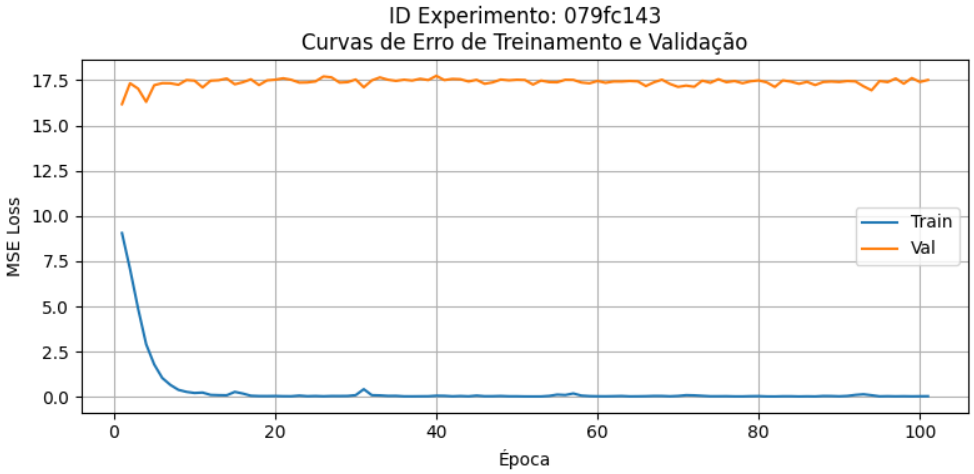


Figura VI.11: Curvas de Aprendizado - Grota Funda

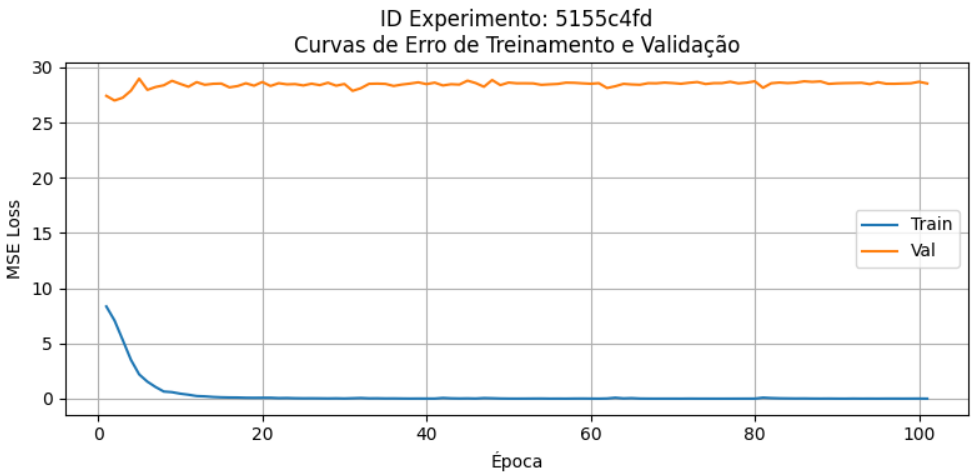


Figura VI.12: Curvas de Aprendizado - Riocentro

VI.2 Análise Retrospectiva

Nesta seção realizamos uma análise retrospectiva sobre os resultados apresentados na Seção VI.1. Iniciamos com análises sobre a distribuição da precipitação nos conjuntos de dados empregados no treinamento dos melhores modelos STGCN. Em seguida analisamos o coeficiente de correlação entre as *features reflectivity* e *precipitation*.

A física da atmosfera é considerada um sistema complexo e caótico. Uma única condição atmosférica inicial pode conduzir a múltiplos cenários. Por conta disso, torna-se um desafio encontrar uma única explicação para os casos em que os resultados foram melhores ou piores. No entanto, detectamos indícios que podem sustentar hipóteses acerca dos resultados e que podem ser explorados em trabalhos futuros.

Dentre as EIs, a estação *Tijuca Muda* foi a que apresentou o melhor resultado geral. O modelo 6, *i.e.*, o modelo treinado com os dados de $\mathcal{N}_6^{(1)}$, apresentou boa convergência de treinamento e bom equilíbrio de acurácia entre as categorias de precipitação. Cabe destacar que o modelo 6 apresentou erro aproximadamente 86% menor que o método IDW. Diante disso, buscamos investigar os motivos para esse bom desempenho e usamos as informações associadas para comparar com as demais EIs.

Também observamos na Figura VI.1a que a acurácia da EI $\mathcal{I}^{(1)}$ tende a aumentar para as categorias *fraca* e *moderada*. De forma inversa, a acurácia tende a diminuir nas categorias *forte* e *extrema*. Partimos de uma primeira hipótese que os dados de vizinhança que são cumulativamente adicionados são responsáveis por esse comportamento. Dessa forma, analisamos os dados da vizinhança $\mathcal{N}_6^{(1)}$ e comparamos com as demais $\mathcal{N}_6^{(i)}$.

A Figura VI.13 apresenta o histograma de precipitação para diferentes conjuntos de dados das estações *Tijuca Muda* e *Campo Grande*. Esse mesmo tipo de histograma é apresentado nos parágrafos seguintes, sempre com a mesma estrutura a seguir. Na primeira linha apresentamos os

conjuntos relacionados ao melhor modelo para *Tijuca Muda*. Na segunda linha apresentamos os conjuntos da estação que está sendo comparada. A primeira coluna apresenta o histograma dos dados das EVs empregados no conjunto de treino. Na segunda coluna, o histograma dos dados das EVs empregados no conjunto de teste. A terceira coluna apresenta o histograma dos dados das EIs, *i.e.*, os dados que devem ser interpolados e que não são empregados nem no treino nem no teste. Os histogramas foram configurados em intervalos de tamanho 5 e estão segmentados por categoria de chuva.

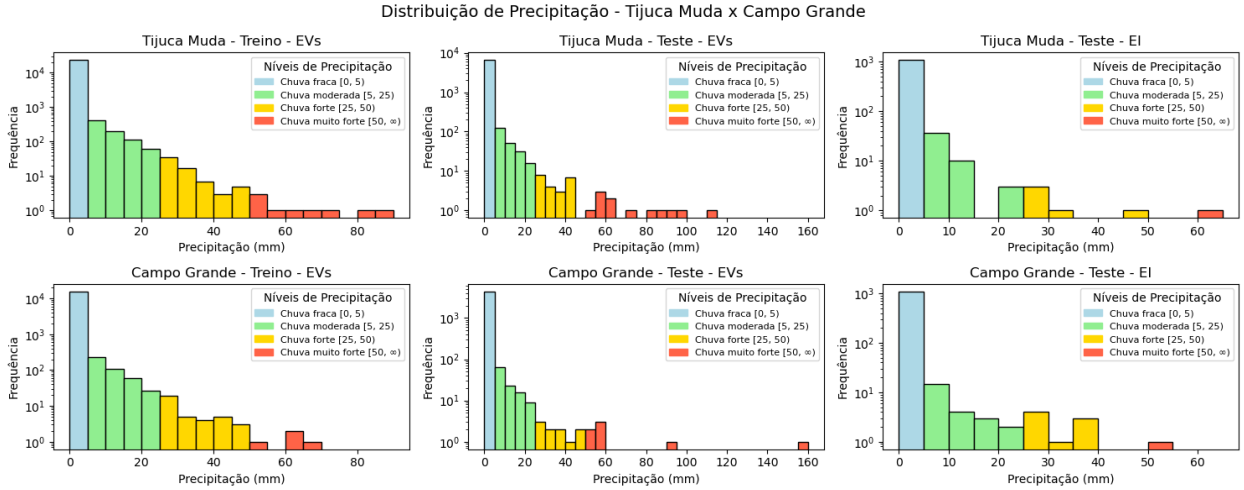


Figura VI.13: Histograma Tijuca Muda x Campo Grande

Observamos na Figura VI.13 que o modelo treinado para a estação *Tijuca Muda* contém mais exemplos de eventos extremos no conjunto de treino, quando comparado ao conjunto de treino $\mathcal{N}_4^{(4)}$ da estação $\mathcal{I}^{(4)}$, *Campo Grande*. Cabe lembrar que a estação $\mathcal{I}^{(4)}$ possui apenas quatro vizinhos. A Figura VI.14 apresenta a mesma comparação, porém com os dados de $\mathcal{N}_6^{(5)}$ da estação $\mathcal{I}^{(5)}$, *Grota Funda*. A quantidade de exemplos de eventos extremos em $\mathcal{N}_6^{(5)}$ é maior do que em $\mathcal{N}_4^{(4)}$ e comparável com a estação *Tijuca Muda*. Apesar disso, todas as estações EVs $\mathcal{N}_6^{(1)}$ de *Tijuca Muda* ficam à menos de $5,2$ km de distância, enquanto que as EVs $4, 5, 6$ de *Campo Grande* ficam à mais de 13 km. Analisando somente as EVs $\mathcal{N}_3^{(5)}$, na Figura VI.15, que distam aproximadamente 8 km, observamos uma quantidade reduzida de eventos extremos no conjunto de treino.

A Figura VI.16 apresenta os histogramas que comparam *Tijuca Muda* com a estação $\mathcal{I}^{(6)}$, *Rio Centro*, empregando as EVs $\mathcal{N}_6^{(6)}$. Novamente temos que as duas estações mais distantes em $\mathcal{N}_6^{(6)}$ estão à $7,7$ km e 10 km aproximadamente. No entanto, o ponto que chama mais atenção são os dados de $\mathcal{N}_6^{(6)}$ no conjunto de teste. O modelo STGCN depende das informações da vizinhança para interpolar a precipitação em $\mathcal{I}^{(6)}$. No entanto, observamos precipitação de mais de 90 mm/h em $\mathcal{I}^{(6)}$, enquanto que conjunto de dados de teste de $\mathcal{N}_6^{(6)}$ apresenta um máximo de 45 mm/h. As duas estações mais próximas em $\mathcal{N}_6^{(6)}$ estão à $4,3$ km e $4,6$ km aproximadamente. É razoável supor que essas estações registrassem volumes maiores de precipitação. Isso conduz à hipótese de

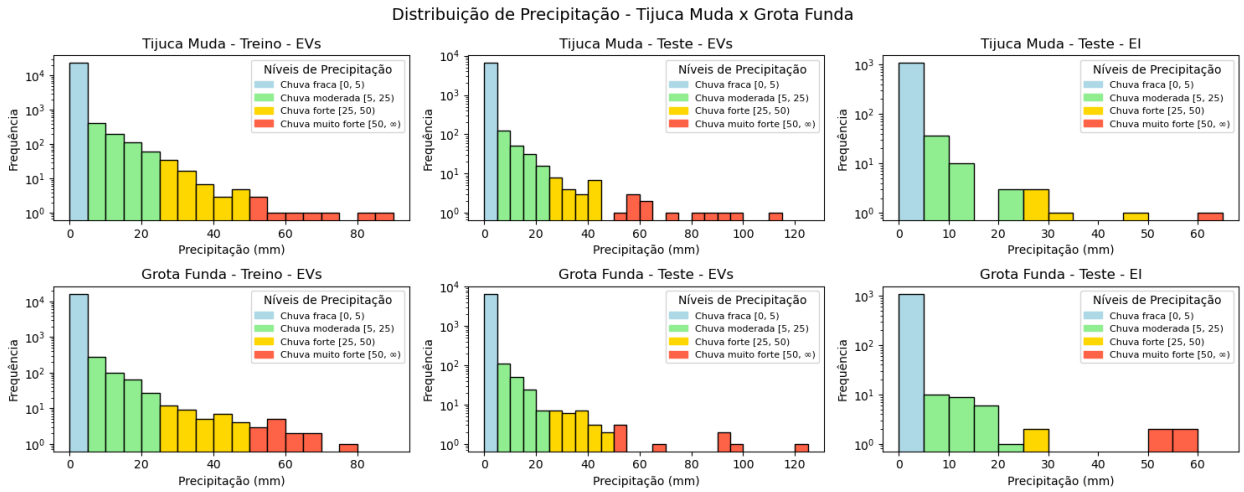


Figura VI.14: Histograma Tijuca Muda x Grota Funda($\mathcal{N}_6$)

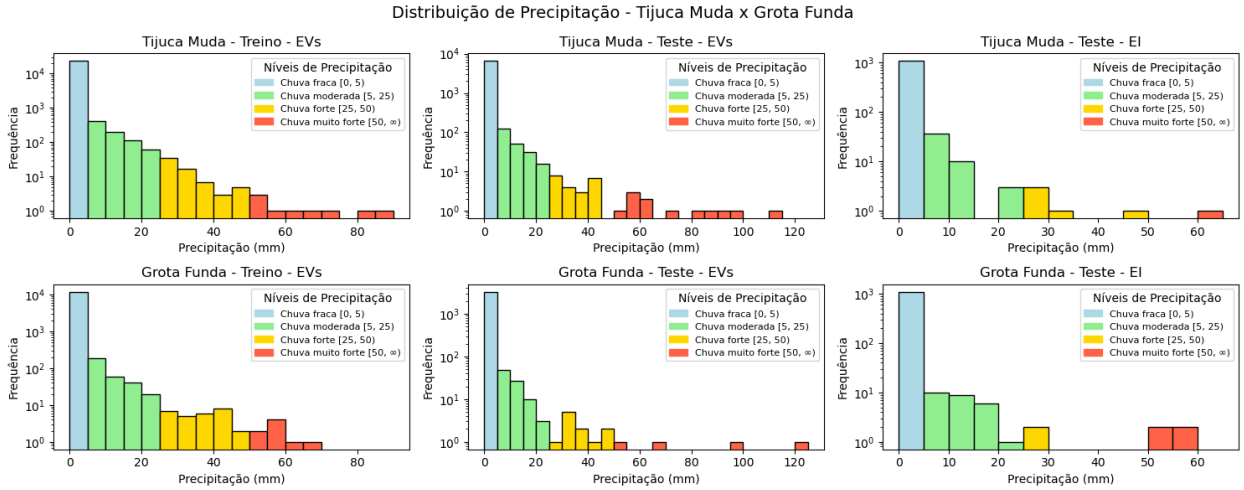


Figura VI.15: Histograma Tijuca Muda x Grota Funda($\mathcal{N}_3$)

que esse subconjunto de dados pode conter inconsistências.

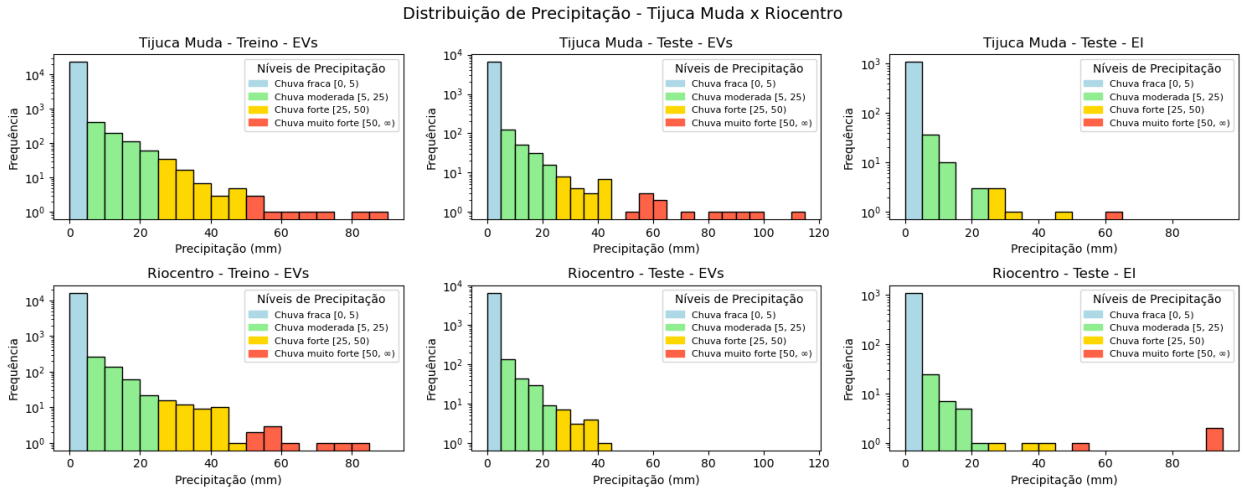


Figura VI.16: Histograma Tijuca Muda x Riocentro

A Figura VI.17 apresenta os histogramas que comparam *Tijuca Muda* com a estação $\mathcal{I}^{(3)}$, *Alto*

da Boa Vista, empregando as EVs $\mathcal{N}_6^{(3)}$. Todas as estações em $\mathcal{N}_6^{(3)}$ estão à menos de $5,7$ km de distância de $\mathcal{I}^{(3)}$. A quantidade de exemplos de eventos extremos nos conjuntos de treino e teste também são comparáveis aos da estação *Tijuca Muda*. Para levantar hipóteses acerca dos motivos do modelo não ter aprendido com os dados de $\mathcal{N}_6^{(3)}$, fizemos uma análise topográfica da região de $\mathcal{I}^{(3)}$ que é apresentada na Figura VI.18.

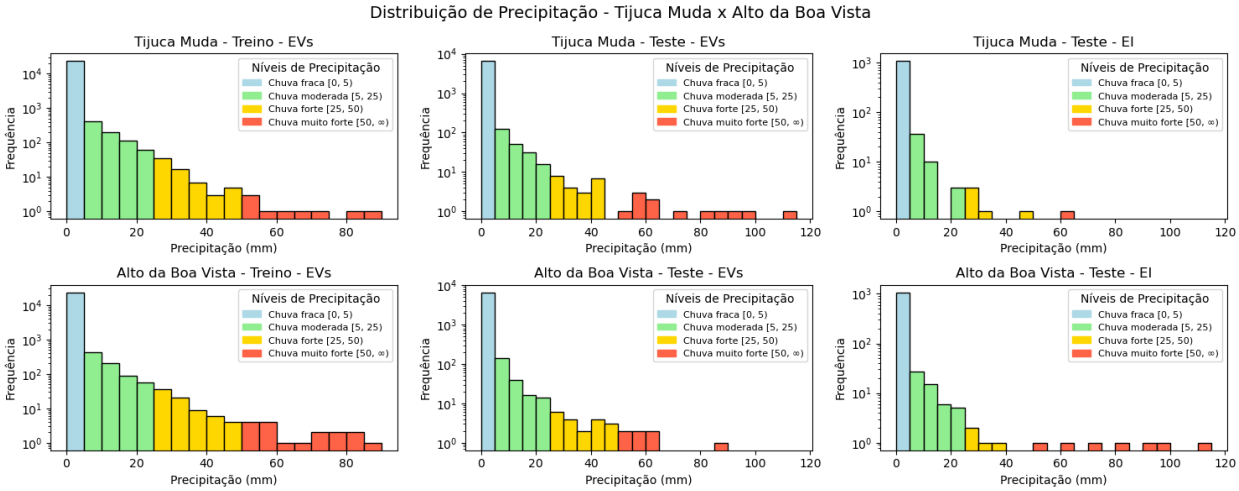


Figura VI.17: Histograma Tijuca Muda x Alto da Boa Vista($\mathcal{N}_6$)



Figura VI.18: Topografia da Região de Alto da Boa Vista

Observamos na Figura VI.18 que as EVs $1, 4, 5, 6$ de $\mathcal{N}_6^{(3)}$ estão do outro lado do *Maciço da Tijuca* em relação à localização de $\mathcal{I}^{(3)}$. O *Maciço da Tijuca* chega a ter 1 km de altitude. É razoável supor que a influência dessas estações sobre $\mathcal{I}^{(3)}$ seja reduzida por conta disso. Na Figura VI.19

analisamos os dados de $\mathcal{N}_3^{(3)}$ que contém as EVs $2, 3$, localizadas do mesmo lado do maciço. Em $\mathcal{N}_3^{(3)}$ observamos quantidade relativamente menor de exemplos de eventos extremos no conjunto de treino. Por fim, a estação $\mathcal{I}^{(3)}$ apresenta valores a serem interpolados relativamente mais altos que os disponíveis nos conjuntos de treino e teste.

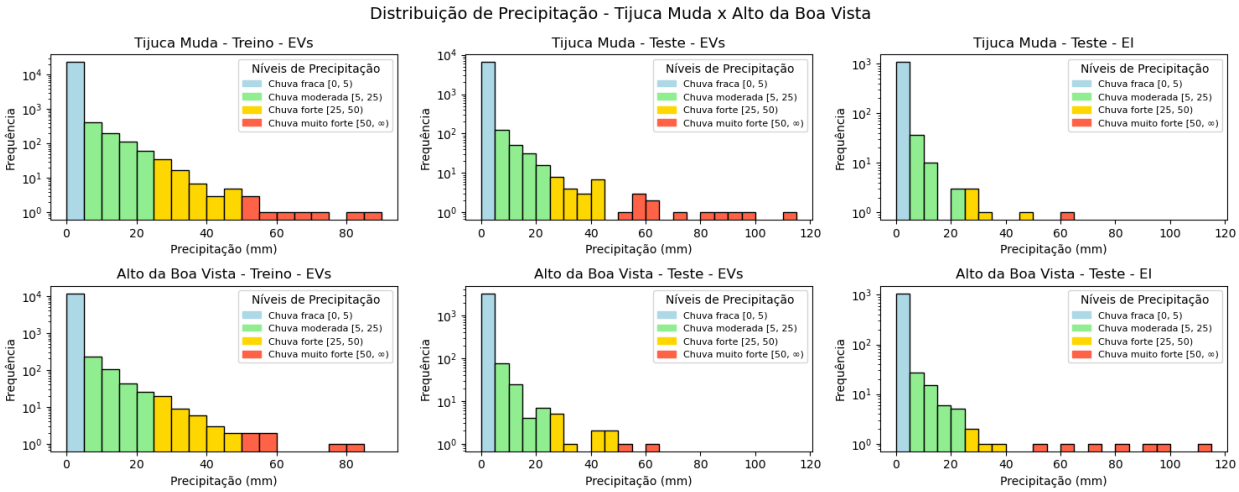


Figura VI.19: Histograma Tijuca Muda x Alto da Boa Vista($\mathcal{N}_3$)

A Figura VI.20 apresenta os histogramas que comparam *Tijuca Muda* com a estação $\mathcal{I}^{(2)}$, *Guaratiba*, empregando as EVs $\mathcal{N}_2^{(2)}$. Cabe lembrar que a estação $\mathcal{I}^{(2)}$ possui apenas duas EVs no raio de 15 km. Observamos mais uma vez uma quantidade relativamente menor de exemplos de eventos extremos em comparação com *Tijuca Muda*. Além disso, um ponto que chama atenção é o valor de precipitação acima de 120 mm/h registrado por $\mathcal{I}^{(2)}$. A EV mais próxima de $\mathcal{I}^{(2)}$ fica à apenas 80 metros de distância, porém em $\mathcal{N}_2^{(2)}$ a máxima registrada fica abaixo de 100 mm/h. Ainda, a EV 2 de $\mathcal{I}^{(2)}$ fica à $8,5$ km de distância, logo o método IDW atribui quase todo o peso à EV 1. No entanto, a Tabela VI.2 apresenta erro alto também para o IDW. Novamente, podemos levantar hipótese de inconsistências no subconjunto de dados.

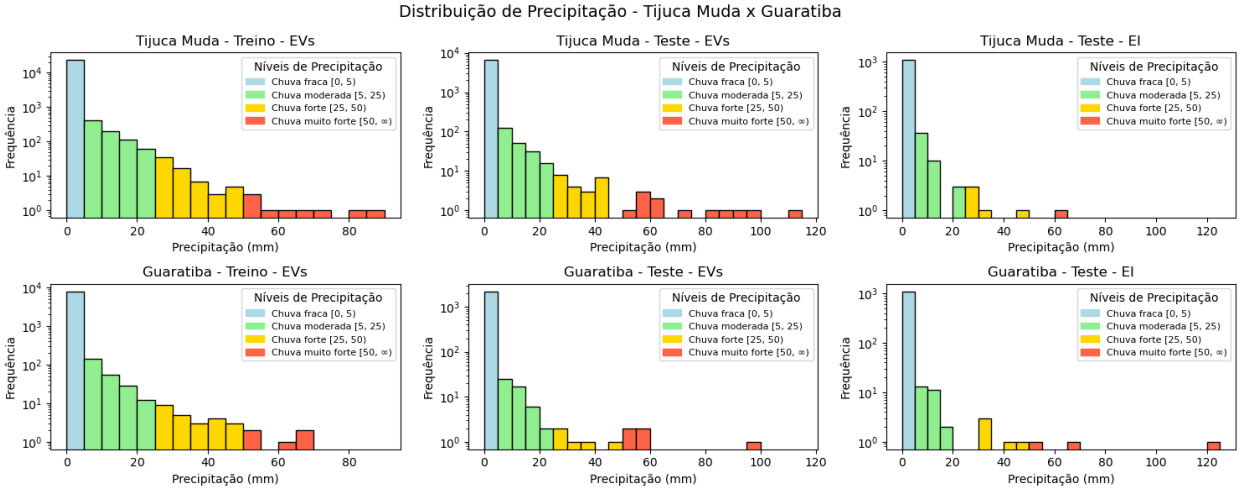


Figura VI.20: Histograma Tijuca Muda x Guaratiba

Para analisar de forma retrospectiva o impacto da *feature reflectivity* sobre a acurácia do modelo, processamos a correlação entre a refletividade e a precipitação nos conjuntos de treino e teste. Conforme detalhado na Seção V.3.2, foram derivadas várias *features* a partir da refletividade observada. Por conta disso, empregamos as *features reflectivity_mean* e *precipitation* para calcular o coeficiente de correlação.

A Figura VI.21 apresenta o coeficiente de correlação para cada EI. Na primeira coluna são apresentados os valores para os dados das EVs no conjunto de treino. A segunda coluna apresenta para os dados das EVs no conjunto de teste. A terceira coluna somente para os dados da EI. Os conjuntos de treino apresentam correlação relativamente menor, o que pode ser um indício de impacto para o aprendizado empregando essa *feature*.

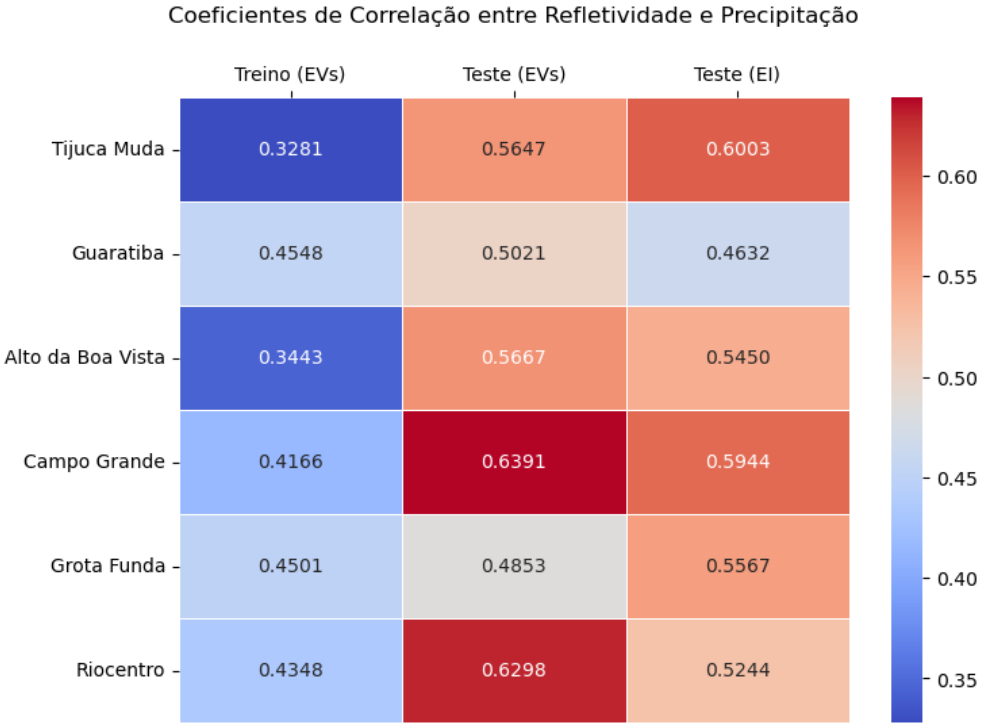


Figura VI.21: Correlação entre Refletividade e Precipitação.

A Figura VI.22 apresenta o coeficiente de correlação separado por ano para cada conjunto de treino. O valor numérico acima de cada barra representa a quantidade de *snapshots* distintos. A linha vermelha representa o coeficiente de correlação para o conjunto de teste. Neste gráfico não é apresentado o conjunto de dados da própria EI.

A correlação varia para cada ano e não segue uma tendência. Também não é observado uma relação direta com a quantidade de *snapshots*. Em geral, o coeficiente de correlação fica abaixo do conjunto de teste para todos os anos, com exceção de *Guaratiba* e *Grota Funda*, ambos no ano de 2020. Isso leva à conclusão de que essa *feature* precisa ser melhor estudada, de forma a transformá-la em um preditor melhor.

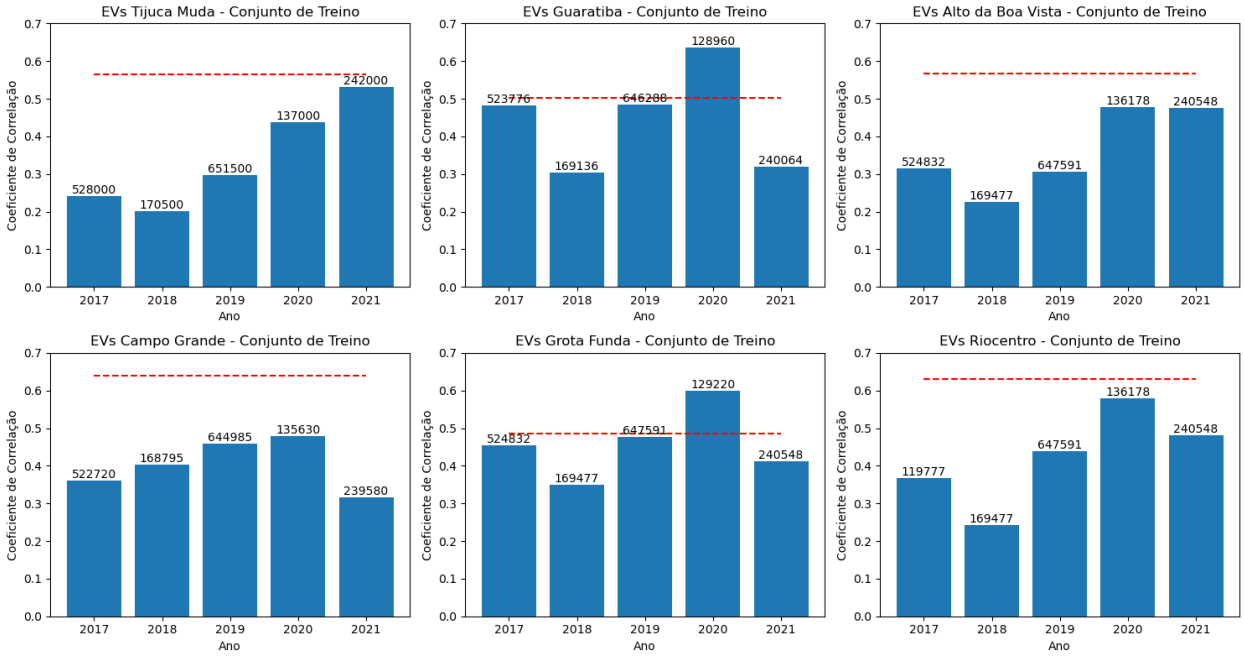


Figura VI.22: Correlação entre Refletividade e Precipitação por Ano.

Capítulo VII Conclusão e Trabalhos Futuros

Este Capítulo apresenta a conclusão sobre o trabalho e discute as possibilidades de trabalhos futuros. A Seção VII.1 resume os objetivos, resultados alcançados e a análise retrospectiva realizada. A Seção VII.2 apresenta sugestões de trabalhos futuros em diferentes abordagens.

VII.1 Conclusão

Este trabalho implementa e avalia a interpolação de dados de precipitação por meio de STGCN. Empregamos séries de dados multivariados de diferentes instrumentos meteorológicos para treinar o modelo. Comparamos a acurácia do modelo STGCN com o método estatístico IDW. Os resultados alcançados apontam que, comparado ao IDW, a STGCN é capaz de obter melhor acurácia em eventos mais extremos de precipitação e alcança resultados promissores em eventos de menor magnitude. Em alguns casos, nosso modelo chega a alcançar uma acurácia para eventos mais extremos entre 40% e 80% maior que o método IDW.

Além da acurácia, avaliamos o modelo sob dois objetivos secundários: *i)* o impacto sobre a acurácia quando empregamos os dados de radar para treinar o modelo e; *ii)* a influência dos dados de *Estações Vizinhas* em uma determinada coordenada de interesse, *i.e.*, o ponto que se deseja interpolar. Identificamos que os dados de radar contribuem positivamente para uma maior acurácia em eventos mais extremos de precipitação. No entanto, isso pode não ocorrer em alguns casos. Analisamos a influência dos dados da vizinhança sobre a acurácia na interpolação de um ponto. Identificamos que a adição gradativa de informação pode convergir o modelo para um bom equilíbrio de acurácia em todas as categorias de precipitação. No entanto, adicionar informação em demasia conduz ao aumento do desbalanceamento dos dados de precipitação, causando viés ao modelo.

Na análise retrospectiva sobre os resultados levantamos hipóteses sobre os motivos que levam o modelo a ter resultados melhores ou piores. Também identificamos indícios de inconsistências nos dados que podem prejudicar o treinamento do modelo STGCN e até mesmo o cálculo do método estatístico IDW. Analisamos a correlação entre os dados de radar e precipitação sob os diferentes conjuntos de treino e teste empregados nos experimentos. Concluímos que a correlação varia muito em diferentes períodos da série histórica, o que conduz à necessidade de um estudo aprofundando

sobre os dados de radar e como empregá-los de forma eficaz no treinamento do modelo.

Este trabalho alcançou os objetivos principal e secundários, além de entregar as contribuições que são: *i)* desenvolvimento de um novo método de interpolação de precipitação baseado em redes neurais para grafos, empregando múltiplas fontes de dados multivariados; *ii)* definição de um processo de *otimização* e *curadoria* para tratamento de dados de diferentes tipos de instrumentos meteorológicos e; *iii)* implantação e disponibilização de um repositório de dados meteorológicos para o projeto *RioNowcast*. Por fim, este trabalho produziu os seguintes artigos publicados:

- F. Porto, "Machine Learning Approaches to Extreme Weather Events Forecast in Urban Areas: Challenges and Initial Results", superfri, vol. 9, no. 1, pp. 49–73, May 2022, https://doi.org/10.14529/jsfi220104.

- A. Fonseca, F. Porto, M. Ferro, E. Ogasawara, and E. Bezerra. "Analysis of precipitation data in Rio de Janeiro city using Extreme Value Theory", in Anais Estendidos do XXXVII Simpósio Brasileiro de Bancos de Dados, Búzios, 2022, pp.193-198, https://doi.org/10.5753/sbbd_estendido.2022.21864.

- Bezerra, Eduardo; Fonseca, Augusto; Cabo, Adriano; Porto, Fabio; Ferro, Mariza. "Precipitation Nowcasting using Data Augmentation", in Anais Estendidos do XXXVIII Simpósio Brasileiro de Bancos de Dados (SBBD), 38. , 2023, Belo Horizonte/MG. p. 318-323. https://doi.org/10.5753/sbbd_estendido.2023.25647.

- Da Fonseca, Augusto J. M.; Freitas, Bruno L.; Bezerra, Eduardo; Bernardini, Flávia C.; Ferro, Mariza. "Diversidade de Dados Meteorológicos da Cidade do Rio de Janeiro: Uma Proposta de Arquitetura de Armazenamento e Fluxo de Dados para Modelos de Previsão", in Anais Estendidos do XXXVIII Simpósio Brasileiro de Bancos de Dados (SBBD), 38. , 2023, Belo HorizonteMG. p. 306-311. https://doi.org/10.5753/sbbd_estendido.2023.235307.

- A. Fonseca, R. Garcia, E. Ogasawara, P. González, F. Porto, and E. Bezerra. "Fusing Satellite and Weather Station Data to Build a Nowcasting Model for Precipitation", in The Latin American Workshop on Information Fusion, Rio de Janeiro, 2023.

- FONSECA, Augusto; GOLDSCHIMIDT, Ronaldo; OGASAWARA, Eduardo; FERRO, Mariza; PORTO, Fábio; BEZERRA, Eduardo. "Interpolação e Previsão de Precipitação por Redes Neurais Convolucionais para Grafos". In: BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB (WEBMEDIA), 30.2024, Juiz de ForaMG. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024 .p.395-399. https://doi.org/10.5753/webmedia.2024.242049.

VII.2 Trabalhos Futuros

Podemos separar as possibilidades de trabalhos futuros em duas abordagens: *data driven* e *model driven*. A primeira baseada em geração e transformação de novas *features* para o treinamento do modelo. A segunda baseada em modificações na arquitetura e hiperparâmetros de treinamento da STGCN. Por fim, podemos realizar experimentos e análises complementares sobre os resultados alcançados neste trabalho.

Na abordagem *data driven* podemos experimentar outras técnicas de transformações de *features*, *e.g.*, distintos métodos de normalização e imputação de dados. Também pretendemos gerar novas *features* que podem enriquecer o contexto espacial dos dados como por exemplo latitude, longitude, altitude, relevo e informação sobre maciços. Também podemos gerar *features* globais que representem eventos sinóticos. Pretendemos experimentar diferentes resoluções temporais e espaciais dos dados, bem como diferentes tamanhos de janela para a série temporal de entrada. Por fim, experimentar a aplicação de *data-augmentation* nos exemplos de eventos mais extremos de precipitação, bem como técnicas de amostragem para diminuir o desbalanceamento.

Na abordagem *model driven* podemos experimentar diferentes hiperparâmetros no treinamento do modelo. Também pretendemos explorar uma função de custo customizada que não seja sensível ao desbalanceamento dos dados. Podemos explorar o treinamento incluindo atributos em arestas, *e.g.*, a distância inversa. Pretendemos experimentar outras arquiteturas STGCN. Por exemplo, arquiteturas que possibilitem o aprendizado em grafos heterogêneos, permitindo modelar cada vértice com diferentes atributos, o que se aproxima do cenário real em que temos distintos instrumentos meteorológicos.

Por fim, podemos realizar novos experimentos e análises com os dados e arquitetura empregados neste trabalho. Pretendemos experimentar e avaliar nosso método em outras coordenadas da área de estudo, avaliando diferentes regiões. Também podemos realizar testes de sensibilidade com outras *features*. Pretendemos analisar os eventos específicos em que o modelo erra mais com o objetivo de encontrar possíveis padrões associados ao erro. Por fim, analisar os resultados por época do ano, investigando padrões sazonais.

Referências Bibliográficas

Huapu Lu, Mingyu Chen, and Wenbo Kuang. The impacts of abnormal weather and natural disasters on transport and strategies for enhancing ability for disaster prevention and mitigation. *Transport Policy*, 98:2–9, 2020. 1

Hossein Azadi, Ali Akbar Barati, Saeedeh Nazari Nooghabi, and Jürgen Scheffran. Climate-related disasters and agricultural land conversion: towards prevention policies. *Climate and Development*, 14(9):814–828, 2022. 1

Peter Hoeppe. Trends in weather related disasters–consequences for insurers and society. *Weather and climate extremes*, 11:70–79, 2016. 1

Steven J. Goodman, Richard J. Blakeslee, William J. Koshak, Douglas Mach, Jeffrey Bailey, Dennis Buechler, Larry Carey, Chris Schultz, Monte Bateman, Eugene McCaul, and Geoffrey Stano. The goes-r geostationary lightning mapper (glm). *Atmospheric Research*, 125-126:34–49, 2013. ISSN 0169-8095. doi: https://doi.org/10.1016/j.atmosres.2013.01.006. URL `https://www.sciencedirect.com/science/article/pii/S0169809513000434`. 1

Xiaoli Ren, Xiaoyong Li, Kaijun Ren, Junqiang Song, Zichen Xu, Kefeng Deng, and Xiang Wang. Deep learning-based weather prediction: A survey. *Big Data Research*, 23:100178, 2021a. ISSN 2214-5796. doi: https://doi.org/10.1016/j.bdr.2020.100178. URL `https://www.sciencedirect.com/science/article/pii/S2214579620300460`. 1

Rafaela Castro, Yania M Souto, Eduardo Ogasawara, Fabio Porto, and Eduardo Bezerra. Stconvs2s: Spatiotemporal convolutional sequence to sequence network for weather forecasting. *Neurocomputing*, 426:285–298, 2021. 1, 4, 25, 26

Gabriela Lewenfus, Wallace Alves Martins, Symeon Chatzinotas, and Björn Ottersten. Joint forecasting and interpolation of graph signals using deep learning. *arXiv preprint arXiv:2006.01536*, 2020. 1, 3, 24, 25, 26

Selim Furkan Tekin, Oguzhan Karaahmetoglu, Fatih Ilhan, Ismail Balaban, and Suleyman Serdar Kozat. Spatio-temporal weather forecasting and attention mechanism on convolutional lstms. *arXiv preprint arXiv:2102.00696*, 2021. 1, 4, 25, 26

IBGE. Rio de Janeiro (RJ) | Cidades e Estados | IBGE. https://www.ibge.gov.br/cidades-e-estados/rj/rio-de-janeiro.html, 2023. 1

IBGE. Populacao_em_areas_de_risco_no_brasil. https://agenciadenoticias.ibge.gov.br/media/com_mediaibge/arquivos/6d4743b1a7387a2f8ede699273970d77.pdf, 2018. 1

Claudine Pereira Dereczynski, Juliana Silva de Oliveira, and Christiane Osório Machado. Climatologia da precipitação no município do rio de janeiro. *Revista Brasileira de Meteorologia*, 24(1): 24–38, 2009. 2

AlertaRio. Quem somos. http://alertario.rio.rj.gov.br/institucional/quem-somos/, Mai 2023. 2

Ryuji Kimura. Numerical weather prediction. *Journal of Wind Engineering and Industrial Aerodynamics*, 90(12):1403–1414, 2002. ISSN 0167-6105. doi: https://doi.org/10.1016/S0167-6105(02)00261-1. URL https://www.sciencedirect.com/science/article/pii/S0167610502002611. Fifth Asia-Pacific Conference on Wind Engineering. 2

Gustavo Bastos Lyra, Tamíres Partelli Correia, José Francisco de Oliveira-Júnior, and Marcelo Zeri. Evaluation of methods of spatial interpolation for monthly rainfall data over the state of rio de janeiro, brazil. *Theoretical and Applied Climatology*, 134(3):955–965, 2018. 2, 3, 11

Brant Liebmann and Dave Allured. Daily precipitation grids for south america. *Bulletin of the American Meteorological Society*, 86(11):1567–1570, 2005. 2, 24

Chris Funk, Pete Peterson, Martin Landsfeld, Diego Pedreros, James Verdin, Shraddhanand Shukla, Gregory Husak, James Rowland, Laura Harrison, Andrew Hoell, et al. The climate hazards infrared precipitation with stations—a new environmental record for monitoring extremes. *Scientific data*, 2(1):1–21, 2015. 2, 24

Hongye Zhou, Feng Zhang, Zhenhong Du, and Renyi Liu. Forecasting pm2. 5 using hybrid graph convolution-based model considering dynamic wind-field to offer the benefit of spatial interpretability. *Environmental Pollution*, 273:116473, 2021. 3, 24, 26

Xuying Yang, Feng Zhang, Peng Sun, Xiaofan Li, Zhenhong Du, and Renyi Liu. A spatio-temporal graph-guided convolutional lstm for tropical cyclones precipitation nowcasting. *Applied Soft Computing*, 124:109003, 2022. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2022.109003. URL https://www.sciencedirect.com/science/article/pii/S1568494622003258. 4

Yunan Qiu, Zhenyu Lu, and Haibo Tang. A short-term regional precipitation prediction model based on wind-improved spatiotemporal convolutional network. *Earth and Space Science*, 9(9):

e2022EA002411, 2022. doi: https://doi.org/10.1029/2022EA002411. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2022EA002411. e2022EA002411 2022EA002411. 4

Minbo Ma, Peng Xie, Fei Teng, Bin Wang, Shenggong Ji, Junbo Zhang, and Tianrui Li. Histgnn: Hierarchical spatio-temporal graph neural network for weather forecasting. *Information Sciences*, 648:119580, 2023. 4

Eric Lehman and Tom Leighton. Mathematics for computer science. https://www.cs.princeton.edu/courses/archive/spring10/cos433/mathcs.pdf, 2004. 12

Stanford. Unsupervised Feature Learning and Deep Learning. http://ufldl.stanford.edu/tutorial/, November 2019. 12, 13, 14

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. 12

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. 12

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019. 17, 22

William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017. 17

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 17, 32

Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I Webb, Irwin King, and Shirui Pan. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 18

Kyunghyun Cho. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 19, 20

Dzmitry Bahdanau. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 20

I.S. Knyazeva, N. G. Makarenko, I. Makarenko, A. Vdovina, and F. A. Urtiev. Analysis of the solar magnetograms evolution using a spectral gap of critical nets. *Journal of Physics: Conference*

Series, 738(1):012070, aug 2016. doi: 10.1088/1742-6596/738/1/012070. URL https://dx.doi.org/10.1088/1742-6596/738/1/012070. 23

Alessio Verdone, Simone Scardapane, and Massimo Panella. Explainable spatio-temporal graph neural networks for multi-site photovoltaic energy production. *Applied Energy*, 353:122151, 2024. 23

Jingjuan Wang, Qingkui Chen, and Huilin Gong. Stmag: A spatial-temporal mixed attention graph-based convolution model for multi-data flow safety prediction. *Information Sciences*, 525: 16–36, 2020. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2020.03.040. URL https://www.sciencedirect.com/science/article/pii/S0020025520302206. 23

Dalin Zhang, Yunjuan Peng, Yumei Zhang, Daohua Wu, Hongwei Wang, and Hailong Zhang. Train time delay prediction for high-speed train dispatching based on spatio-temporal graph convolutional network. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):2434–2444, 2022. doi: 10.1109/TITS.2021.3097064. 23

Hao Zhang, Chunyue Song, Jie Zhang, Hui Wang, and Jinlong Guo. A multi-step airport delay prediction model based on spatial-temporal correlation and auxiliary features. *IET Intelligent Transport Systems*, 15(7):916–928, 2021. doi: https://doi.org/10.1049/itr2.12071. URL https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/itr2.12071. 23

Junkai Sun, Junbo Zhang, Qiaofei Li, Xiuwen Yi, Yuxuan Liang, and Yu Zheng. Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(5):2348–2359, 2020. 24

Zhiqiang Lv, Jianbo Li, Chuanhao Dong, and Zhihao Xu. Deepstf: A deep spatial–temporal forecast model of taxi flow. *The Computer Journal*, 66(3):565–580, 11 2021. ISSN 0010-4620. doi: 10.1093/comjnl/bxab178. URL https://doi.org/10.1093/comjnl/bxab178. 24

Fan Hou, Yue Zhang, Xinli Fu, Lele Jiao, and Wen Zheng. The prediction of multistep traffic flow based on ast-gcn-lstm. *Journal of Advanced Transportation*, 2021(1):9513170, 2021. doi: https://doi.org/10.1155/2021/9513170. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/9513170. 24

Junjie Zhou, Siyue Shuai, Lingyun Wang, Kaifeng Yu, Xiangjie Kong, Zuhua Xu, and Zhijiang Shao. Lane-level traffic flow prediction with heterogeneous data and dynamic graphs. *Applied Sciences*, 12(11), 2022. ISSN 2076-3417. doi: 10.3390/app12115340. URL https://www.mdpi.com/2076-3417/12/11/5340. 24

Zhi Liu, Jixin Bian, Deju Zhang, Yang Chen, Guojiang Shen, and Xiangjie Kong. Dynamic multi-view coupled graph convolution network for urban travel demand forecasting. *Electronics*, 11 (16), 2022. ISSN 2079-9292. doi: 10.3390/electronics11162620. URL https://www.mdpi.com/2079-9292/11/16/2620. 24

Linmu Zou, Zijia Wang, and Rui Guo. Real-time prediction of transit origin–destination flows during underground incidents. *Transportation Research Part C: Emerging Technologies*, 163: 104622, 2024. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2024.104622. URL https://www.sciencedirect.com/science/article/pii/S0968090X24001438. 24

Hamed Farahmand, Yuanchang Xu, and Ali Mostafavi. A spatial–temporal graph deep learning model for urban flood nowcasting leveraging heterogeneous community features. *Scientific Reports*, 13(1):6768, 2023. 24

Yumeng Wang, Ke Liu, Yuejun He, Pengfei Wang, Yuxin Chen, Hang Xue, Caiyi Huang, and Lin Li. Enhancing air quality forecasting: A novel spatio-temporal model integrating graph convolution and multi-head attention mechanism. *Atmosphere*, 15(4), 2024. ISSN 2073-4433. doi: 10.3390/atmos15040418. URL https://www.mdpi.com/2073-4433/15/4/418. 24

Xiaoli Ren, Xiaoyong Li, Kaijun Ren, Junqiang Song, Zichen Xu, Kefeng Deng, and Xiang Wang. Deep learning-based weather prediction: A survey. *Big Data Research*, 23:100178, 2021b. 24

Jiandong Bai, Jiawei Zhu, Yujiao Song, Ling Zhao, Zhixiang Hou, Ronghua Du, and Haifeng Li. A3t-gcn: Attention temporal graph convolutional network for traffic forecasting. *ISPRS International Journal of Geo-Information*, 10(7):485, 2021. 32

Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, Guzman Lopez, Nicolas Collignon, and Rik Sarkar. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, page 4564–4573, 2021. 32

Apêndice A Estações Vizinhas

Tabela A.1: Tijuca Muda - Estações Vizinhas

Id	Tipo	Distância (km)	Latitude	Longitude
1	Pluviômetro	2.220095	-22.931940	-43.221670
2	Pluviômetro	2.739652	-22.922220	-43.267500
3	Meteorológica	4.587330	-22.896670	-43.221670
4	Pluviômetro	4.808559	-22.931670	-43.196390
5	Meteorológica	4.872877	-22.972780	-43.223890
6	Meteorológica	5.133136	-22.965830	-43.278330
7	Pluviômetro	5.782264	-22.940560	-43.187500
8	Pluviômetro	5.890067	-22.890560	-43.278060
9	Pluviômetro	5.901368	-22.985830	-43.245000
10	Meteorológica	6.723293	-22.992500	-43.233060
11	Pluviômetro	6.996295	-22.896060	-43.187860

Tabela A.2: Guaratiba - Estações Vizinhas

Id	Tipo	Distância (km)	Latitude	Longitude
1	Meteorológica	0.085492	-23.050278	-43.595556
2	Pluviômetro	8.496525	-23.014440	-43.521390

Tabela A.3: Alto da Boa Vista - Estações Vizinhas

Id	Tipo	Distância (km)	Latitude	Longitude
1	Pluviômetro	4.072871	-22.985830	-43.245000
2	Pluviômetro	4.974400	-22.922220	-43.267500
3	Pluviômetro	5.133136	-22.932780	-43.243330
4	Pluviômetro	5.221541	-23.008490	-43.299650
5	Meteorológica	5.501979	-22.992500	-43.233060
6	Meteorológica	5.626820	-22.972780	-43.223890
7	Pluviômetro	5.898770	-22.925560	-43.315830
8	Pluviômetro	6.918093	-22.931940	-43.221670
9	Pluviômetro	8.369688	-22.890560	-43.278060
10	Pluviômetro	8.847459	-22.891820	-43.310050
11	Pluviômetro	8.935688	-22.945560	-43.362780
12	Pluviômetro	9.209953	-22.931670	-43.196390
13	Meteorológica	9.327617	-22.988333	-43.190556
14	Pluviômetro	9.382768	-22.986390	-43.189440
15	Meteorológica	9.633679	-22.896670	-43.221670
16	Pluviômetro	9.715381	-22.940560	-43.187500
17	Pluviômetro	10.650127	-22.912500	-43.364720
18	Pluviômetro	11.486274	-22.955830	-43.166670
19	Meteorológica	11.659997	-22.977210	-43.391550
20	Pluviômetro	12.010873	-22.873330	-43.338890
21	Pluviômetro	12.084068	-22.896060	-43.187860
22	Meteorológica	13.062036	-22.940000	-43.402778
23	Pluviômetro	13.501567	-22.844440	-43.275280

Tabela A.4: Campo Grande - Estações Vizinhas

Id	Tipo	Distância (km)	Latitude	Longitude
1	Pluviômetro	5.611089	-22.856940	-43.541110
2	Pluviômetro	10.181284	-22.880280	-43.465830
3	Meteorológica	12.563947	-22.909440	-43.684440
4	Pluviômetro	13.004293	-23.014440	-43.521390

Tabela A.5: Grota Funda - Estações Vizinhas

Id	Tipo	Distância (km)	Latitude	Longitude
1	Pluviômetro	8.287362	-23.010000	-43.440560
2	Meteorológica	8.496525	-23.050280	-43.594720
3	Meteorológica	8.572017	-23.050278	-43.595556
4	Pluviômetro	13.004293	-22.903610	-43.561940
5	Meteorológica	13.920073	-22.977210	-43.391550
6	Meteorológica	14.695532	-22.940000	-43.402778

Tabela A.6: Riocentro - Estações Vizinhas

Id	Tipo	Distância (km)	Latitude	Longitude
1	Meteorológica	4.294294	-22.940000	-43.402778
2	Pluviômetro	4.589359	-22.945560	-43.362780
3	Pluviômetro	6.201692	-23.010000	-43.440560
4	Pluviômetro	7.702070	-22.912500	-43.364720
5	Pluviômetro	9.648616	-22.925560	-43.315830
6	Pluviômetro	10.029393	-23.008490	-43.299650
7	Meteorológica	11.659997	-22.965830	-43.278330
8	Pluviômetro	12.641586	-22.891820	-43.310050
9	Pluviômetro	12.747886	-22.873330	-43.338890
10	Meteorológica	13.038013	-22.861389	-43.411389
11	Pluviômetro	13.192193	-22.880280	-43.465830
12	Pluviômetro	13.920073	-23.014440	-43.521390
13	Pluviômetro	14.097064	-22.922220	-43.267500