UFRPE

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

Avaliação de Redes Adversárias Generativas para Imputação Múltipla de Séries Temporais Meteorológicas

Wellington Luiz Antonio

Recife - PE, Agosto 2023

Wellington Luiz Antonio

Avaliação de Redes Adversárias Generativas para Imputação Múltipla de Séries Temporais Meteorológicas

Dissertação submetida à Coordenação do Programa de Pós-Graduação em Informática Aplicada do Departamento de Estatística e Informática - DEINFO - Universidade Federal Rural de Pernambuco, como parte dos requisitos necessários para obtenção do grau de Mestre.

Universidade Federal Rural de Pernambuco – UFPE

Departamento de Estatística e Informática

Programa de Pós-Graduação em Informática Aplicada

Orientador: Prof. Dr. Glauco Estácio Gonçalves

Coorientador: Prof. Dr. Victor Wanderley Costa de Medeiros

Recife - PE

Agosto 2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

A635a Antonio, Wellington Luiz
Avaliação de redes adversárias generativas para imputação múltipla de séries temporais meteorológicas / Wellington Luiz Antonio. - 2023.
62 f. : il.

Orientador: Glauco Estacio Goncalves.
Coorientador: Victor Wanderley Costa de Medeiros.
Inclui referências.

Dissertação (Mestrado) - Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Informática Aplicada, Recife, 2023.

1. aprendizagem de máquina. 2. inteligência artificial. 3. IA Generativa. 4. meteorologia. I. Goncalves, Glauco Estacio, orient. II. Medeiros, Victor Wanderley Costa de, coorient. III. Título

CDD 004

Wellington Luiz Antonio

Avaliação de Redes Adversárias Generativas para Imputação Múltipla de Séries Temporais Meteorológicas

Dissertação submetida à Coordenação do Programa de Pós-Graduação em Informática Aplicada do Departamento de Estatística e Informática - DEINFO - Universidade Federal Rural de Pernambuco, como parte dos requisitos necessários para obtenção do grau de Mestre.

Aprovada em Recife - PE, Agosto 30, 2023:

Prof. Dr. Glauco Estácio Gonçalves
(Orientador)
Universidade Federal do Pará

Prof. Dr. Cícero Garrozi
Universidade Federal Rural de Pernambuco

Prof. Dr. Rodrigo Gabriel Ferreira Soares
Universidade Federal Rural de Pernambuco

Recife - PE
Agosto 2023

Dedico a Aristides Luis Antonio (in memoriam)

Agradecimentos

Agradeço ao Deus trino, que sempre me protegeu e me indicou os melhores caminhos a serem seguidos para chegar até onde estou, eu, um aluno de escola pública que viveu toda sua infância, adolescência e parte da juventude criando gado para sustento da família, nunca poderia imaginar que um dia estaria concluindo um mestrado, algo que desde criança me chamou a atenção. Foi esse Deus maravilhoso que me concedeu esse sonho.

Agradeço aos meus queridos e estimados orientadores (Prof. Dr. Glauco Estácio Gonçalves e Prof. Dr. Victor Wanderley Costa de Medeiros) por serem exemplos a serem seguidos tanto como professores quanto pesquisadores. Agradeço-vos pelas orientações que me foram dadas no Juá Labs, desde o início da pesquisa até a conclusão desse trabalho. Essas orientações e ensinamentos me tornaram um profissional melhor e foram fundamentais para que eu conseguisse concluir esse trabalho, sou eternamente grato por tudo que fizeram e estão fazendo por mim.

À Neurotech, especialmente a todos da Squad 3 (220V) que vem sempre me apoiando nas adversidades da vida pessoal/profissional/acadêmica, principalmente ao coordenador João Carlos e nosso gestor Luciano Oliveira que me deram maior apoio para conseguir alguns dias para finalizar essa dissertação. Desde já, o meu muito obrigado.

À minha querida e amada esposa, Aline Roberta, que me ajuda em todo tempo, mesmo com todos os compromissos das disciplinas do seu curso superior e do estágio/trabalho, sempre me apoiou nos momentos difíceis deste mestrado e principalmente nesses últimos meses. Amo você, sou grato a Deus e a você por fazer parte da minha vida.

À minha família, principalmente meus pais, que mesmo sem saber exatamente o que faço, me apoiam. Meu pai (que faleceu durante desenvolvimento deste trabalho) e minha mãe que nunca imaginou que teria um filho que iria fazer um curso superior e principalmente uma pós-graduação em uma universidade pública federal, até hoje ela à chama de escola. Amo vocês, sou grato por terem me criado e por me ajudarem nas mais variadas situações dessa vida.

A todos da igreja Assembleia de Deus no povoado Angélicas – Vicência – PE, que sempre me apoiaram e encorajaram a crescer no conhecimento e sabedoria, sou grato por toda a paciência e orações para comigo.

Aos que fazem parte do Laboratório em Infraestrutura Computacional – Juá labs, especialmente aos meus amigos queridos, José Clodoalves e Diego Bezerra, que sempre estão dispostos a me ajudar, obrigado por me ajudarem neste trabalho. Sou grato por vocês e pelo Juá labs, pois, esse laboratório é de grande relevância quando se trata de

aprendizado e geração de conhecimento.

Por fim, agradeço aos que participaram e me ajudaram direta ou indiretamente na minha vida acadêmica e pessoal durante todo o decorrer do meu percurso na graduação.

"O temor do Senhor é o princípio da sabedoria"
(Provérbios 9,10a)

Resumo

A ausência de dados (lacunas), decorrente de problemas ocorridos durante a coleta, processamento e armazenamento de informações é um desafio constante na análise de séries temporais meteorológicas. Este problema, caso não seja tratado de maneira apropriada, pode impactar a eficácia dos métodos de previsão, sendo uma ameaça à qualidade e confiabilidade de qualquer análise. Neste contexto, a imputação de dados meteorológicos assume um papel de destaque, havendo diferentes métodos disponíveis na literatura. Recentemente, destaca-se o método GAIN (*Generative Adversarial Imputation Nets*), proposto por Yoon, Jordon e Van Der Schaar em 2018, que promove o uso de Redes Adversárias Generativas (*Generative Adversarial Networks*) com o intuito de imputar dados ausentes, conforme a distribuição de probabilidade aprendida a partir de dados presentes. Para a análise do GAIN, particionou-se o conjunto de dados de séries temporais de temperatura máxima das estações meteorológicas da APAC (Agência Pernambucana de Águas e Clima) em: treino (80%), validação (10%) e teste (10%). No conjunto de treino e validação foram inseridos 5% de lacunas artificialmente para melhor controle do experimento. Na validação, utilizou-se a ferramenta *Random Search* para escolha dos melhores hiperparâmetros e dentro dessa ferramenta uma variação da técnica de validação cruzada, *K-Fold*, foi adotada com o objetivo de aumentar a confiabilidade do modelo comparando-se 2 redes GAIN, uma dita rede GAIN de base e outra rede GAIN aprimorada. Essas duas últimas tiveram seu desepenho comparado aos métodos de imputação KNN (*K-Nearest Neighbours*) e MICE (*Multiple Imputation by Chained Equations*). Na fase de teste os algoritmos em apreço nesse estudo foram testados à variação de lacunas de 5 à 50%. Para medir o desempenho destes frente a variação no percentual de lacunas, utilizou-se as métricas MAE (*Mean Absolute Error*) e RMSE (*Root Mean Squared Error*). Os resultados obtidos demonstram que o KNN e MICE, embora mais simples apresentaram melhor desempenho nas métricas supracitadas do que o GAIN. Apesar disso, este último demonstrou robustez frente ao treinamento com lacunas permanecendo estável, não apresentando variações significativas com o aumento do percentual de lacunas. Um outro achado, foi a melhoria do desempenho do algoritmo GAIN com a proposição de uma rede GAIN aprimorada através do uso do método *Random Search* para escolha de melhores hiperparâmetros de entrada e adição de mais uma camada oculta. Através dessas modificações observou-se uma melhoria no desempenho do GAIN, mas ainda inferior aos modelos tradicionais de imputação, demonstrando que a depender de cada contexto e características específicas do conjunto de dados, deve-se ter a sensibilidade para escolher o método de imputação mais adequado às características do problema.

Palavras-chaves: aprendizagem de máquina, inteligência artificial, IA Generativa, meteorologia.

Abstract

The absence of data (gaps), resulting from issues during data collection, processing, and storage, is a constant challenge in the analysis of meteorological time series. If not handled properly, this problem can impact the effectiveness of forecasting methods, posing a threat to the quality and reliability of any analysis. In this context, the imputation of meteorological data plays a prominent role, with various methods available in the literature. Recently, the GAIN method (Generative Adversarial Imputation Nets), proposed by Yoon, Jordon, and Van Der Schaar in 2018, stands out. It employs Generative Adversarial Networks to impute missing data based on the learned probability distribution from available data. For the analysis of GAIN, the time series dataset of maximum temperature from APAC (Pernambuco Agency for Water and Climate) meteorological stations was partitioned into training (80%), validation (10%), and test (10%) sets. In the training and validation sets, 5% of gaps were artificially inserted for better experiment control. In the validation phase, the Random Search tool was used to select the best hyperparameters, and within this tool, a variation of the K-Fold cross-validation technique was adopted to enhance the model's reliability, comparing two GAIN networks, one referred to as the base GAIN and the other as an enhanced GAIN. These two were compared to imputation methods KNN (K-Nearest Neighbors) and MICE (Multiple Imputation by Chained Equations). In the testing phase, the algorithms considered in this study were tested with gap variations from 5% to 50%. To measure their performance in response to gap percentage variation, the metrics MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error) were employed. The results obtained show that KNN and MICE, although simpler, outperformed GAIN in the mentioned metrics. Nonetheless, GAIN demonstrated robustness in the face of training with gaps, remaining stable and not showing significant variations as the gap percentage increased. Another finding was the improved performance of the GAIN algorithm with the introduction of an enhanced GAIN network using the Random Search method for selecting better input hyperparameters and the addition of an extra hidden layer. These modifications resulted in improved GAIN performance but still fell short of traditional imputation models, emphasizing the importance of sensitivity in selecting the most appropriate imputation method based on the specific dataset's characteristics and context.

Keywords: machine learning, artificial intelligence, Generative AI, Meteorology.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AIC	Critério de Informação de Akaik (*Akaike Information Criterion*)
APAC	Agência Pernambucana de Águas e Clima
API	*Application Programming Interface*
ARMA	*Autoregressive Moving Average*
BIC	Critério de Informação Bayesiano (*Bayesian Information Criterion*)
CNN	*Convolutional Neural Network*
DSS	Sistemas de Suporte a Decisão
GAIN	*Generative Adversarial Imputation Nets*
IC	*Informative Censoring*
IoT	Internet das Coisas (*Internet of Things*)
KNN	*K-Nearest Neighbors*
MF	*Matrix Factorization*
MAE	*Mean Absolute Error*
MAR	*Missing at Random*
MCAR	*Missing Completely at Random*
MICE	*Multiple Imputation by Chained Equations*
MNAR	*Not Missing at Random*
MSE	*Mean Squared Error*
PNI	*Pattern Non-Ignorable*
RMSE	*Root Mean Squared Error*
RNN	*Recurrent Neural Network*
SB	*Selection Bias*
UFRPE	Universidade Federal Rural de Pernambuco

Sumário

1 Introdução

A realização de previsões que envolvam variáveis climáticas é muito comum e útil no campo da meteorologia, onde faz-se uso de modelos preditivos cada vez mais aprimorados. Contudo, para realização destas previsões uma quantidade considerável de dados de estações meteorológicas é necessária, mas nem sempre essas bases de dados com séries temporais apresentam um conjunto de dados completo, tendo em vista questões adversas relacionadas a coleta, armazenamento e transferência desses dados, como por exemplo um sensor na estação terrestre com mau funcionamento ou até mesmo uma placa de rede de um servidor de banco de dados, onde os dados meteorológicos ficam armazenados (FLORES; TITO; CENTTY, 2019; BEZERRA et al., 2019).

Observa-se, portanto que, a presença de lacunas em conjuntos de dados é um problema comum que afeta a qualidade e utilidade das análises de dados e previsões. Alguns problemas decorrentes dessa ausência de dados incluem vieses em resultados de análise, uma vez que as amostras incompletas podem não ser representativas do verdadeiro comportamento dos dados em análise bem como poderá ter efeitos na precisão e confiabilidade desses modelos (MIR et al., 2022).

A ausência de dados pode levar à perda de informações valiosas, especialmente se as lacunas ocorrerem em momentos críticos ou em pontos importantes do conjunto de dados afetando diretamente a variável de interesse. Além do que a falta de dados pode levar a estimativas menos precisas de condições meteorológicas, por exemplo, o que pode afetar negativamente a qualidade de previsões (FLORES; TITO; SILVA, 2019).

Dessa forma, lacunas precisam ser preenchidas ou tratadas para evitar erros em previsões e demais aplicações em ciência de dados. Nesse contexto, a imputação de dados surge como uma solução a este problema. A utilização de métodos de imputação como o KNN, MICE e GAIN pode ajudar a preencher lacunas e minimizar os problemas associados aos dados ausentes (YOON; JORDON; SCHAAR, 2018; DIOUF; DÈME et al., 2022).

Neste trabalho buscou-se apresentar e avaliar o desempenho e robustez do método de imputação de dados GAIN, para preenchimento de lacunas em conjuntos de dados meteorológicos do estado de Pernambuco, cedidos pela APAC (Agência Pernambucana de Águas e climas). Buscou-se também realizar uma análise comparativa com algoritmos tradicionais em termos de métricas que serão explanadas ao longo deste estudo, implementando-se contribuições para melhoria do desempenho do algoritmo GAIN diante de cenários adversos, ou seja, com a presença de variados percentuais de lacunas artificialmente inseridos no conjunto de dados.

1.1 Motivação

Em muitos cenários da vida cotidiana, é comum encontrar lacunas nos dados devido a vários motivos, como falhas na coleta, erros de medição ou outros fatores. A presença desses valores ausentes pode comprometer a qualidade das análises estatísticas, modelagem preditiva e outras tarefas de processamento de dados.

A imputação de dados é um processo de preenchimento de valores ausentes com estimativas razoáveis, a fim de manter a integridade dos dados e permitir uma análise mais completa e precisa. No contexto de séries temporais, a imputação de dados é especialmente relevante devido à natureza temporal dos dados e às implicações que lacunas ou valores ausentes podem ter na análise dessas séries.

As séries temporais são conjuntos de observações coletadas em intervalos regulares ao longo do tempo, e são comuns em diversas áreas, como economia, meteorologia, finanças e monitoramento de processos industriais. Desta forma, alguns pontos impulsionaram este trabalho no uso da imputação de dados em séries temporais para dados meteorológicos, a saber (POPOLIZIO et al., 2021; JING et al., 2022; DIOUF; DÈME et al., 2022; LUO et al., 2019; SAMAL et al., 2021):

- **Preservação de Padrões Temporais**: A presença de valores ausentes pode quebrar padrões temporais importantes nas séries, afetando a capacidade de realizar análises precisas e de fazer previsões.
- **Melhoria na Precisão das Previsões**: Muitos modelos de previsão exigem séries temporais completas para fornecer estimativas precisas do futuro. A imputação permite que esses modelos sejam aplicados com mais confiança.
- **Manutenção de Consistência Temporal**: Valores ausentes podem resultar em inconsistências ao calcular médias, somas ou outras métricas temporais, afetando a interpretação dos resultados.
- **Validação de Hipóteses**: Análises baseadas em séries temporais frequentemente envolvem a testagem de hipóteses sobre tendências, sazonalidades e efeitos temporais. A imputação ajuda a manter a integridade dessas análises.
- ***Completeness* dos Dados**: Dados completos ao longo do tempo são essenciais para entender eventos passados, identificar padrões sazonais e detectar mudanças de longo prazo.
- **Melhoria na Eficiência do Modelo**: A imputação pode resultar em modelos mais eficientes e estáveis, uma vez que a presença de dados completos permite a aplicação de algoritmos mais avançados.

- **Redução de *Bias***: Valores ausentes podem introduzir viés nos resultados das análises de séries temporais, afetando a interpretação dos padrões e tendências.

Além dos pontos já citados, outro ponto importante é o uso de um algoritmo de ponta, que utiliza redes adversária generativas para a imputação de dados meteorológicos em séries temporais utilizando dados da APAC. Este algoritmo tem se destacado por sua eficácia e desempenho no campo da imputação de dados. Sua abordagem inovadora, baseada em redes neurais adversárias, demonstra uma notável desenvoltura na resolução do desafio de lidar com valores ausentes em conjuntos de dados.

1.2 Objetivos

O principal objetivo deste trabalho é avaliar o comportamento de redes adversárias generativas para imputação (GAIN) de dados de temperatura multivariados e espacialmente distribuídos.

E ainda como objetivos específicos, pretende-se:

- Avaliar o impacto da quantidade de lacunas no conjunto de treinamento na eficácia do método GAIN;
- Investigar o comportamento do GAIN frente a outros algoritmos clássicos da literatura para imputação de dados meteorológicos;

1.3 Organização do trabalho

Este trabalho está organizado em seis capítulos, sendo o Capítulo 1 de introdução e o Capítulo 2 sobre referencial teórico de imputação de dados meteorológicos em series temporais, os algoritmos e as métricas utilizadas. O Capítulo 3 apresenta os trabalhos relacionados. O Capítulo 4 descreve os materiais utilizados para a coleta, armazenamento e pré-processamento dos dados da APAC para os experimentos, além disso, tem-se também os métodos utilizados para a avaliação das redes adversárias para a imputação nos conjuntos de dados com lacunas e ainda neste capítulo é falado sobre a técnica de otimização de hiperparâmetros, *Random Seach*. No Capítulo 5, são apresentados e discutidos os resultados obtidos. Já no Capítulo 6, são apresentadas as conclusões e considerações finais deste trabalho, além de destacar sugestões para pesquisas futuras.

2 Referencial Teórico

Neste capítulo serão apresentados conceitos gerais necessários ao entendimento dessa dissertação.

2.1 Imputação de dados meteorológicos em séries temporais

As séries temporais são modelos comumente utilizados para a representação de dados do mundo físico. Estas são objeto de estudo de diversos ramos, como: medicina, mercado financeiro, previsões meteorológicas, dentre outros (YANG et al., 2020).

Séries temporais são utilizadas para descrição do comportamento de dados ao longo do tempo. No caso de dados meteorológicos, por exemplo, estes são frequentemente coletados em intervalos de tempo regulares, a cada hora ou a cada dia para serem utilizados na construção dessas séries temporais.

Contudo, durante a aferição de dados meteorológicos, lacunas em registros de equipamentos de captação de dados são comuns. O aparecimento dessas lacunas geralmente está relacionado a condições externas do ambiente onde essa captação ocorre. Algumas das condições externas que podem incidir no aparecimento de lacunas no conjunto de dados são: falhas técnicas, manutenção inadequada, acesso restrito a áreas remotas, mudanças nas tecnologias de coleta de dados ou mesmo desastres naturais (JING et al., 2022; MIR et al., 2022).

Quanto às falhas técnicas tem-se as falhas ocorridas em sensores de captação ou demais instrumentos de coleta de dados ou equipamentos de armazenamento. A falta de manutenção adequada nos dispositivos de coleta de dados pode também levar a falhas e lacunas nos conjuntos de dados. É importante realizar inspeções regulares, calibração e substituição de equipamentos conforme necessário para garantir a precisão contínua dos dados. Existem também casos de ausência de dados em regiões de difícil acesso devido a condições geográficas desafiadoras, como terrenos montanhosos, florestas densas ou regiões polares. Isso pode dificultar a implantação e manutenção de dispositivos de coleta de dados, resultando em lacunas na cobertura dessas áreas.

À medida que a tecnologia avança, podem ocorrer mudanças nas metodologias de coleta de dados. Se houverem transições de dispositivos ou métodos de medição, podem haver lacunas temporárias nos conjuntos de dados enquanto a transição é feita. Outro caso que resulta no aparecimento de lacunas nos conjuntos de dados são os eventos climáticos extremos, como tempestades, furacões, incêndios florestais ou terremotos, podendo interromper ou destruir sistemas de coleta.

Uma solução que vem sendo comumente utilizada para esse problema é a imputação de dados em séries temporais. Esta ação representa uma atividade que contribui significativamente para a homogeneização dos dados, pois é por meio desta que séries temporais subsequentes podem ser utilizadas em previsões (FLORES; TITO; CENTTY, 2020). Esta técnica será melhor detalhada no próximo subtópico.

2.1.1 Imputação

A imputação é uma técnica estatística utilizada para lidar com dados faltantes em conjuntos de dados. A falta de dados pode ser causada por uma variedade de razões, incluindo erros de medição, falhas na coleta de dados ou exclusões intencionais de dados (JING et al., 2022). Dentro desse contexto, existe a imputação de dados meteorológicos que é uma técnica utilizada para preenchimento de lacunas em séries temporais.

A imputação de dados consiste, portanto, em uma técnica que substitui as lacunas em conjunto de dados de séries temporais por valores estimados por meio de diferentes métodos (ELY et al., 2021)

Em séries históricas de dados, um fato importante a se destacar é a existência da relação de dependência entre os dados coletados e o tempo registrado na série temporal, por isso, a imputação de dados adequada proporciona um conjunto de dados mais completo para implementação de modelos e uma melhor análise da experimentação (YANG et al., 2020) e (MIR et al., 2022).

A imputação de dados é amplamente utilizada na análise climática e meteorológica, pois dados incompletos podem prejudicar a precisão das análises e previsões (FLORES; TITO; CENTTY, 2020). A imputação de dados é um processo em que os valores ausentes em um conjunto de dados são estimados ou preenchidos com base em determinados métodos ou modelos. Dentro desse conjunto de métodos e modelos existe duas abordagens principais para a imputação de dados: imputação simples e imputação múltipla (NUNES; KLÜCK; FACHEL, 2010).

A imputação simples é definida como uma técnica em que os valores ausentes são substituídos por uma única estimativa que pode ser o valor médio ou um valor interpolado de observações próximas (LITTLE; RUBIN, 2019; EMMANUEL et al., 2021).

Uma das abordagens mais simples de imputação é a imputação por média, em que os valores ausentes são substituídos pela média dos valores observados no mesmo dia ou na mesma hora do dia. Essa técnica é simples e fácil de aplicar, especialmente em séries temporais de dados meteorológicos.

No entanto, a imputação por média também apresenta algumas limitações e pode não ser adequada para todos os casos. Por exemplo, se a série temporal tiver grandes flutuações ou mudanças abruptas, a imputação por média pode não fornecer uma estimativa

precisa dos valores ausentes. Além disso, a imputação por média pode produzir uma estimativa enviesada se os dados ausentes estiverem correlacionados com outras variáveis da série temporal.

Além da imputação simples tem-se também a múltipla que trata-se de um método avançado para lidar com dados ausentes, que oferece uma abordagem mais completa em comparação com a imputação simples. Nesse método, múltiplas séries de imputação são geradas por meio de cálculos de imputação repetidamente, com o objetivo de refletir a incerteza associada aos dados ausentes (RUBIN, 1987; LITTLE; RUBIN, 2019).

Primeiramente, são criadas várias séries de imputação, cada uma delas preenchendo os valores ausentes de forma estocástica. Isso significa que valores aleatórios são introduzidos nas posições dos dados ausentes, com base em uma distribuição adequada que leve em consideração as informações disponíveis nos dados observados. A geração de múltiplas séries de imputação é importante para capturar a variabilidade inerente aos dados ausentes e fornecer estimativas mais robustas.

Em seguida, os parâmetros estatísticos de interesse são estimados separadamente para cada série de imputação. Isso pode ser feito usando os métodos estatísticos tradicionais aplicados aos dados completos, levando em consideração as observações tanto imputadas quanto não imputadas. Essas estimativas são obtidas para cada série de imputação, criando assim uma distribuição de estimativas que reflete a incerteza associada à imputação (RUBIN, 1987; LITTLE; RUBIN, 2019).

Finalmente, critérios de avaliação são empregados para selecionar a imputação ótima ou combinar as estimativas obtidas a partir das séries de imputação. Esses critérios podem ser baseados em medidas de erro, como o erro quadrático médio ou o erro relativo, ou em critérios de informação, como o Critério de Informação de Akaike (AIC) ou o Critério de Informação Bayesiano (BIC). A imputação ótima é aquela que minimiza o critério de avaliação escolhido.

A imputação múltipla se enquadra no tipo de lacunas de dados MCAR, que será melhor explícita na subseção 2.1.2. Nesse tipo de lacuna, a probabilidade de um dado estar ausente é completamente aleatória e não depende dos valores observados ou não observados. No caso da imputação múltipla, os valores ausentes são estimados por meio da geração de múltiplas séries de imputação. Cada série de imputação é gerada introduzindo valores aleatórios nas posições dos dados ausentes, com base em uma distribuição adequada e considerando as informações disponíveis nos dados observados (BLEIDORN et al., 2022).

A imputação múltipla assume que os dados ausentes estão faltando de forma aleatória e, portanto, segue o mecanismo MCAR. Isso significa que a probabilidade de um dado estar ausente não está relacionada aos valores reais ou não reais desse dado, mas é determinada exclusivamente por fatores aleatórios desconhecidos. É importante

notar que, a imputação múltipla também pode ser aplicada em outros tipos de lacunas de dados, como o MAR e MNAR, onde o tipo MAR representa àquelas que não são explicadas por ela mesma, mas por outra variável presente no conjunto de dados. Já a MNAR representa as lacunas que são explicadas pela própria variável em questão ou por outras que necessariamente não estejam presentes no conjunto de dados (RUBIN, 1976).

Ainda dentro do contexto da imputação múltipla, podemos fazer uma distinção entre imputação univariada e imputação multivariada. Na imputação univariada, os valores ausentes de uma variável são estimados com base em dados da própria variável de maneira que a imputação seja realizada tratando-se a variável de forma independente das demais, ou seja, sem considerar relações com outras variáveis (LITTLE; RUBIN, 2019).

Por outro lado, na imputação multivariada, os valores ausentes são estimados levando-se em conta a relação entre as várias variáveis do conjunto de dados. Nesse caso, as variáveis observadas são utilizadas para imputar os valores ausentes em todas as variáveis simultaneamente, levando em consideração a estrutura multivariada dos dados (ENDERS, 2022; ZHANG et al., 2021; LITTLE; RUBIN, 2019).

A utilização da imputação univariada e imputação multivariada depende da natureza dos dados e do objetivo da análise. A imputação univariada pode ser mais adequada quando as variáveis são independentes ou possuem uma relação simples entre si. Já a imputação multivariada é útil quando as variáveis estão inter-relacionadas e compartilham informações relevantes para a imputação.

Para se obter uma imputação mais assertiva e poder assim manipular os dados obtidos de maneira adequada, é importante também entender os tipos de lacunas existentes. Estas subdividem-se em: MCAR, MAR e MNAR (RUBIN, 1976). Esta discussão será abordada na subseção a seguir.

2.1.2 Tipos de lacunas nos dados

A teoria dos tipos de lacunas nos dados foi introduzida originalmente por Rubin. Para introduzir os tipos de lacunas de dados uma matriz representativa $\boldsymbol{R}$ de lacunas em conjuntos de dados que assume valores binários adaptada do trabalho de Emmanuel et al. pode ser definida pela expressão 2.1 que se segue.

$$R = \begin{cases} 1, & \text{se } Y_i \text{ é observado} \\ 0, & \text{se } Y_i \text{ é lacuna} \end{cases} \tag{2.1}$$

Os elementos $\boldsymbol{Y}$ na posição $\boldsymbol{i}$ dessa matriz são classificados de acordo com a presença ou ausência de lacunas. Quando não há a presença de lacunas, ou seja, o valor de $\boldsymbol{Yi}$ é observado, então este assume o valor de 1 e quando há presença de lacunas, atribui-se

então o valor 0.

Os três tipos de lacunas mais comumente apresentadas na literatura são: MCAR, MAR e MNAR (RUBIN, 1976). *Missing Completely at Random* (MCAR) consiste na probabilidade de uma lacuna depender de uma outra variável do conjunto de dados ou da própria variável. Quando os dados são classificados como MCAR, isso significa que as lacunas nestes são realmente aleatórias e não há padrões ou razões subjacentes para a sua ocorrência.

Considerando-se, por exemplo, um conjunto de dados de temperatura máxima em diferentes cidades ao longo de um período de tempo, se alguns desses dados de temperatura estiverem faltantes devido a falhas de transmissão ou registro, e a probabilidade desses dados faltarem não depender de nenhuma variável, como a própria temperatura ou a localização da cidade, então esses dados são considerados faltantes completamente aleatórios (MCAR).

Já o segundo tipo de lacuna considerada na literatura é a *Missing at Random* (MAR), que representa a probabilidade de uma lacuna depender de quaisquer variáveis do conjunto de dados, com exceção da própria variável. Isso significa que a probabilidade de um valor estar faltando pode depender de outras variáveis que estão presentes no conjunto de dados (RUBIN, 1976).

Assim, por exemplo, considerando-se um conjunto de dados com lacunas de temperatura máxima para duas cidades, teria-se que a probabilidade de um dado estar faltando dependeria apenas da temperatura da outra cidade no mesmo dia, mas não da própria temperatura da cidade em questão. A informação da correlação entre as variáveis precisa ser conhecida previamente para se conseguir imputar esse tipo de lacuna (MAR).

O terceiro tipo de lacuna definido na literatura é o *Missing Not at Random* (MNAR), que descreve a probabilidade de uma lacuna depender da própria variável ou de outras variáveis não observadas no conjunto de dados. Isso significa que a falta de um valor está relacionada com os valores que este poderia ter assumido (RUBIN, 1976).

Em um conjunto de dados de temperatura máxima para uma cidade específica, por exemplo, supondo que a probabilidade de um dado estar faltando dependa da própria temperatura da cidade em questão, se em dias muito quentes a temperatura máxima não for registrada, as lacunas nos dados serão mais propensas a ocorrer em dias quentes, independentemente da temperatura de outras cidades. Neste caso, as lacunas nos dados são consideradas não aleatórias (MNAR).

Algumas outras definições e classificações de lacunas de dados são propostas na literatura, como a *Missingness Mechanism Classification Framework* (MMCF) de Little e Rubin. Eles propuseram neste *Framework* seis tipos de lacunas, incluindo os três tipos definidos por Rubin.

Pattern Non-Ignorable (PNI) trata-se de um tipo de lacuna que se classifica como tal caso a probabilidade de uma observação estar faltando dependa do valor dessa observação, além das variáveis observadas. Nesse caso, a probabilidade de lacunas é uma função das variáveis observadas e não observadas, bem como dos valores das observações faltantes.

Um exemplo desse tipo de lacuna pode ser identificado no caso em que ao coletar-se informações sobre a temperatura máxima em uma região específica, em dias extremamente quentes, pode ocorrer que os termômetros dessa estação meteorológica fiquem sobrecarregados devido ao superaquecimento promovendo assim o aparecimento de lacunas nos registros de temperaturas com valores abaixo do real. Nesse caso, a probabilidade de uma lacuna ocorrer depende do próprio valor observado. Assim, se a temperatura máxima registrada em um determinado dia for 50 graus Celsius, é mais provável que uma lacuna ocorra para temperaturas abaixo desses 50 graus do que acima dele. Logo, no PNI, a probabilidade de uma lacuna ocorrer dependerá do valor observado.

Outro tipo de lacuna é o *Selection Bias* (SB). Nesse caso, as lacunas são afetadas por meio de seleção devido a um processo de amostragem que não é aleatório. Assim, a probabilidade de uma observação estar faltando dependerá da probabilidade de seleção (LITTLE; RUBIN, 2019). Assim, ao obter-se informações sobre a temperatura máxima em uma região onde há muitas áreas urbanas com asfalto e concreto, estas áreas podem estar mais quentes do que as áreas rurais devido ao efeito das ilhas de calor. Logo, se a coleta de informações ocorrer apenas nas áreas urbanas, haverá um viés de seleção na amostragem das informações, pois as informações coletadas não seriam representativas da população inteira. Desse modo, a probabilidade de uma lacuna dependerá da probabilidade de seleção.

Já no tipo *Informative Censoring* (IC), as lacunas nos dados são devido a um processo de censura informativa, ou seja, a lacuna é devido a uma observação que está abaixo ou acima de um determinado valor e é afetada pela informação que está faltando. Nesse caso, a probabilidade de aparecer lacunas dependerá do valor da observação que está censurada (LITTLE; RUBIN, 2019). Nesse caso, em uma coleta de informações sobre a temperatura máxima durante um período de tempo específico onde ocorram tempestades, danificando ou desligando a estação meteorológica, o resultado será a presença de lacunas nos registros. Nesse caso, a informação sobre a lacuna é afetada pela observação que está acima de um determinado valor, ou seja, a temperatura máxima antes do evento climático.

Em resumo, lacunas em conjunto de dados podem surgir por diversos motivos, desde problemas técnicos durante a coleta, falta de resposta do respondente ou impossibilidade de medir uma determinada variável. Dessa forma, conforme o Framework proposto por Rubin, observou-se que as lacunas podem ser classificadas principalmente em três tipos: MAR, MCAR e MNAR e que a partir desses três tipos, outros tipos de lacunas podem ser surgir, como foi o caso das dos tipos PNI, SB e IC, explicitadas.

Em geral, a presença de lacunas nos dados pode afetar a validade das análises estatísticas e, por isso, é fundamental lidar com as lacunas de maneira cuidadosa e rigorosa e a imputação de dados mostra-se como uma saída para este problema. Nesse sentido, algoritmos para imputação de dados tem sido desenvolvidos e utilizados na literatura como solução para preenchimento de lacunas em conjunto de dados, em especial dados meteorológicos, fonte de estudo do referido trabalho. Na seção 2.2, serão abordados alguns dos algoritmos que vem sendo utilizados para imputação. (JING et al., 2022; YANG et al., 2020; MIR et al., 2022).

2.2 Algoritmos de imputação de dados meteorológicos

Nesta seção será abordado a literatura acerca dos algoritmos utilizados neste trabalho, KNN (*K-Nearest Neighbors*), MICE e GAIN. Os algoritmos KNN e MICE são os clássicos e mais utilizados, já o GAIN é um dos algoritmos mais recentes, que vem demonstrando bons resultado nas imputações realizadas de acordo com o estado da arte levantado.

2.2.1 KNN – *K-Nearest Neighbors*

O *K-Nearest Neighbors* (KNN) é um algoritmo de aprendizagem de máquina amplamente utilizado para problemas de classificação e regressão. Ele é considerado um algoritmo de aprendizado supervisionado e baseado em instâncias (FERRERO et al., 2008).

O funcionamento do KNN é relativamente simples: dado um novo exemplo de entrada (um ponto), o algoritmo encontra os k vizinhos (outros pontos) mais próximos a ele baseado em uma distância (geralmente a distância euclidiana) e atribui a classe ou valor de saída com base na classe da maioria dos vizinhos ou nos valores desses vizinhos. A escolha do valor de hiperparâmetro k afeta diretamente o desempenho do algoritmo, pois um valor muito baixo pode levar a uma classificação ou regressão instável, enquanto um valor muito alto pode introduzir viés na previsão (DIOUF; DÈME et al., 2022; FERRERO et al., 2008).

O processo utilizado pelo KNN pode ser dividido em 4 etapas: distância, identificação dos k vizinhos mais próximos, agregação dos valores dos vizinhos e estimação do valor do modelo de teste. O KNN é um algoritmo de aprendizado de máquina do tipo "preguiçoso" *(lazy learning)*, o que significa que ele não segue o processo tradicional de treinamento dos modelos de aprendizado de máquina. Em vez disso, o KNN armazena os exemplos de treinamento em uma estrutura de dados para que possam ser consultados rapidamente durante a fase de teste (DIOUF; DÈME et al., 2022).

A primeira etapa é caracterizada pela determinação da medida de distância necessária para calcular a proximidade entre o modelo de treinamento e o modelo de teste. A distância euclidiana é a medida comumente utilizada no KNN, mas também é possível utilizar outras medidas de distância, como a distância de Manhattan ou a de Minkowski.

Com a medida de distância e o valor de k definidos, o algoritmo encontra os k modelos de treinamento mais próximos do modelo de teste com base na distância calculada. Esses modelos então tornam-se os vizinhos mais próximos.

Na terceira etapa (agregação dos valores dos vizinhos), para realizar a regressão, é necessário determinar um valor único para o modelo de teste com base nos valores dos vizinhos mais próximos. A fórmula do KNN para regressão calcula a média dos valores dos vizinhos, atribuindo esse valor como a estimativa para o modelo de teste.

Por fim, o valor estimado para o modelo de teste é obtido como a média dos valores dos vizinhos mais próximos. Esse valor é, portanto, considerado como a resposta do modelo de regressão.

Em suma, a principal premissa do KNN é que pontos de dados semelhantes tendem a estar próximos uns dos outros no espaço de características. Portanto, ao atribuir uma classe ou prever um valor, o KNN considera a maioria das classes dos vizinhos mais próximos (para classificação) ou calcula a média dos valores dos vizinhos (para regressão) (MORAES; ARRAES, 2012).

A fórmula do KNN para regressão pode ser representada da seguinte maneira:

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i, \text{ onde } y_i \text{ são os valores das k observações mais próximas} \tag{2.2}$$

Na formula 2.2, $\hat{y}$ representa o valor estimado de teste. Os valores y_i, são os valores dos k vizinhos mais próximos. A fórmula calcula a média dos valores dos vizinhos, atribuindo esse valor como a estimativa para o exemplo de teste (DIOUF; DÈME et al., 2022).

Embora o KNN seja comumente aplicado em problemas de classificação e regressão em conjuntos de dados tabulares, ele também pode ser usado para realizar imputação de lacunas de dados em séries temporais (KIANI; SALEEM, 2017; LUO et al., 2019; SAMAL et al., 2021). Para um melhor entendimento do funcionamento do KNN tem-se o o algoritmo 1 para imputação.

No algoritmo 1 um passo a passo da execução do KNN é mostrado. O algoritmo inicia uma função para cálculo da distância euclidiana entre dois pontos, $\boldsymbol{a}$ e $\boldsymbol{b}$, por exemplo. Para isso o algoritmo verifica cada linha do conjunto de dados em busca da presença de lacunas. Caso a posição $\boldsymbol{i}$ possua lacuna, o cálculo da distância desta lacuna aos demais pontos do conjunto de dados é iniciado. A fórmula utilizada para o cálculo da distância *nan* euclidiana encontra-se expressa na Equação 2.3, onde $\boldsymbol{w}$ representa o peso de cada

Algoritmo 1 KNN (K-*Nearest Neighbors*) para imputação.

Função CalcularDistancia(a, b)
    Calcular a distância entre a e b
    Retornar a distância calculada
Fim Função
Função KNNImputer(*dados*, k)
    Para cada linha i em *dados* **Faça**
        Se i tiver lacunas de dados **Então**
            Calcular as distâncias entre i e todas as outras linhas em *dados*
            Classificar as linhas com base nas distâncias e selecionar as k linhas mais próximas
            Calcular a média dos valores das k linhas mais próximas
            Substituir os valores ausentes na linha i pela média calculada
        Fim Se
    Fim Para
    Retornar dados atualizados
Fim Função

par de coordenadas, obtido pela razão entre número total de coordenadas pelo número de coordenadas atuais. Nessa expressão a variável $\boldsymbol{d}$ representa a distância entre cada par de pontos.

$$D_{ab} = \sqrt{w * [d_b - d_a]^2} \tag{2.3}$$

Ao final do cálculo das distâncias entre todos os pontos do conjunto de dados, o algoritmo classifica as linhas com base nas distâncias e seleciona as $\boldsymbol{k}$ linhas mais próximas para calcular a média entre elas. Com a média calculada, os valores ausentes da linha i são substituídos e o algoritmo retorna os dados atualizados.

2.2.2 MICE - *Multiple Imputation by Chained Equations*

O algoritmo MICE - *Multiple Imputation by Chained Equations* é caracterizado como um método de imputação múltipla por meio equações em cadeia (MICE), uma vez que mais de uma variável possui dados ausentes, não havendo portanto uma definição padrão dos dados faltantes. Além disso, o MICE realiza a imputação de dados faltantes de cada variável ajustada por meio de modelos de regressão linear (OLIVEIRA et al., 2020).

O método foi proposto por (RUBIN; SCHENKER, 1986) e tem como objetivo realizar a geração de sucessivos modelos de regressão, onde cada variável que apresente lacunas pode ser modelada em relação às demais variáveis do banco de dados.

A ideia básica do algoritmo apreciado nesta seção é imputar valores ausentes em um conjunto de dados, modelando cada variável ausente condicionalmente às variáveis observadas. O processo envolve, portanto, imputar iterativamente os valores ausentes,

utilizando modelos estatísticos para prever os valores faltantes com base nas informações disponíveis nas variáveis observadas. Essa imputação é realizada repetidamente em várias iterações, permitindo a geração de múltiplas imputações (OLIVEIRA et al., 2020).

Após a imputação dos valores ausentes, os resultados das análises estatísticas são combinados usando regras específicas para levar em consideração a variabilidade entre as imputações. Essa combinação dos resultados leva em conta a incerteza introduzida pela imputação dos dados ausentes e permite uma análise estatística mais robusta (OLIVEIRA et al., 2020).

Nos modelos de regressão linear denomina-se de $\mathbf{x}$ a variável a ser imputada na lacuna. As variáveis que apresentam lacunas ($\mathbf{x}$) são identificadas no modelo de imputação como variável dependente e as demais variáveis entram como variáveis independentes (OLIVEIRA et al., 2020; DIOUF; DÈME et al., 2022). Para realização da imputação de valores faltantes em séries temporais, utilizando o MICE, o processo utilizado ocorre em 4 passos: marcação de posição, imputação por regressão, geração de múltiplas imputações para posterior fase de combinação (DIOUF; DÈME et al., 2022; OLIVEIRA et al., 2020).

O primeiro passo realiza uma imputação simples da média para cada valor ausente no conjunto de dados. Nesse passo, substitui-se temporariamente os valores ausentes pelo valor médio da variável correspondente, chamado de marcação de posição. Essas imputações médias servem como estimativas iniciais antes de acontecer o refinamento das imputações usando modelos de regressão.

Uma vez que os marcadores de posição estão no lugar, o segundo passo envolve a aplicação de modelos de regressão para cada variável com valores faltantes. Cada variável faltante é tratada como a variável de resposta em um modelo de regressão, enquanto as outras variáveis observadas servem como preditoras. O objetivo é estabelecer relações estatísticas entre as variáveis e usar essas relações para prever os valores faltantes com base nos valores observados.

Após a realização da imputação por regressão, segue o terceiro passo, que é a geração de múltiplas imputações para cada valor faltante. Isso é feito introduzindo um componente aleatório nos modelos de regressão. A incerteza inerente à imputação é capturada por esse componente aleatório, resultando em diferentes valores imputados para a mesma observação faltante. O número de imputações pode ser definido previamente e influencia a precisão do processo de imputação.

O último passo envolve a combinação das múltiplas imputações geradas nas iterações anteriores. A média das imputações pode ser calculada para cada valor faltante, proporcionando uma única estimativa que leva em consideração a variabilidade introduzida pelo componente aleatório. Além disso, é possível calcular os intervalos de confiança para as estimativas, considerando a distribuição das imputações (OLIVEIRA et al., 2020; RUBIN,

1976).

Para uma maior percepção de como funciona do MICE tem-se o algoritmo 2 que demonstra o seu pseudocódigo.

Os passos de 2 a 4 são repetidos para cada uma das variáveis com valores ausentes, a fim de atualizar todas as imputações usando os modelos de regressão. Esse processo é realizado iterativamente para melhorar as estimativas imputadas.

A equação do MICE para regressão é usada na etapa de imputação de lacunas do algoritmo. Essa equação permite estimar os valores ausentes de uma variável alvo (Y) com base nas demais variáveis explicativas (X_1, X_2, ..., X_k) disponíveis no conjunto de dados.

A equação do MICE para regressão é definida no formato da equação 2.4.

$$Y_{\text{lacuna}}^{(m)} = \beta_0^{(m)} + \beta_1^{(m)} X_1 + \beta_2^{(m)} X_2 + \ldots + \beta_k^{(m)} X_k + \epsilon^{(m)} \tag{2.4}$$

Na Equação 2.4, $Y_{\text{lacuna}}^{(m)}$ representa o valor imputado para a variável ($\mathbf{Y}$) com lacuna na iteração m do algoritmo MICE. Os coeficientes de regressão são representados por $\beta_0^{(m)}, \beta_1^{(m)}, \beta_2^{(m)}, ..., \beta_k^{(m)}$, onde $\beta_0^{(m)}$ é o intercepto e $\beta_1^{(m)}, \beta_2^{(m)}, ..., \beta_k^{(m)}$ são os coeficientes correspondentes às variáveis explicativas (X_1, X_2, ..., X_k). O termo $\epsilon^{(m)}$ representa o erro aleatório que incorpora a incerteza da imputação (OLIVEIRA et al., 2020; RUBIN, 1976).

A equação do MICE para regressão é uma ferramenta fundamental no algoritmo MICE, permitindo a imputação de valores ausentes por meio de modelos de regressão que consideram as relações entre as variáveis disponíveis.

A Equação 2.4 é aplicada em cada iteração do MICE, através da utilização dos valores observados das variáveis explicativas e as imputações atuais das variáveis alvo e explicativas. Ela captura as relações lineares entre as variáveis e permite gerar múltiplas imputações que refletem as características dos dados.

Ao repetir o processo MICE por várias iterações, as imputações são refinadas e atualizadas com base nas estimativas dos modelos de regressão. Isso ajuda a reduzir a incerteza e melhorar a qualidade das imputações, resultando em um conjunto completo de dados imputados que podem ser utilizados para análises subsequentes.

2.2.3 GAIN - *Generative Adversarial Imputation Nets*

O algoritmo GAIN, ou *Generative Adversarial Imputation Nets*, é um método de imputação de dados ausentes que utiliza conceitos da teoria dos jogos e redes neurais generativas adversárias para gerar estimativas precisas e realistas dos valores faltantes (YOON; JORDON; SCHAAR, 2018).

O GAIN é baseado nos princípios do GAN (*Generative Adversarial Network*). Este

Algoritmo 2 MICE (*Multiple Imputation by Chained Equations*)

Função MICE($\mathbf{X}$, M, I)
  Entrada: Conjunto de dados incompletos $\mathbf{X}$, número de múltiplas imputações M, número de iterações I
  Saída: Conjunto de dados imputados $\mathbf{X}_{\text{imputados}}$
  $\mathbf{X}_{\text{imputados}} \leftarrow$ Inicialize com valores iniciais ou médias
  Para m de 1 até M **Faça**
    Para i de 1 até I **Faça**
      Para cada valor da posição j em $\mathbf{X}$ com valores ausentes **Faça**
        $\mathbf{Y} \leftarrow \mathbf{X}_{\text{imputados}}$ exceto a coluna j
        $\mathbf{Y}_j \leftarrow \mathbf{X}[:, j]$
        Impute os valores ausentes em $\mathbf{Y}_j$ com um modelo de regressão estatístico baseado nas variáveis em $\mathbf{Y}$
        Atualize a coluna j em $\mathbf{X}_{\text{imputados}}$ com os valores imputados
      Fim Para
    Fim Para
    Guarde a imputação atual em $\mathbf{X}^{(m)}_{\text{imputados}}$
  Fim Para
  Retornar as múltiplas imputações em $\mathbf{X}^{(1)}_{\text{imputados}}, \mathbf{X}^{(2)}_{\text{imputados}}, \ldots, \mathbf{X}^{(M)}_{\text{imputados}}$
Fim Função

trata-se de um tipo de modelo generativo que consiste em dois componentes principais: o Gerador e o Discriminador. O Gerador aprende a gerar amostras sintéticas que se assemelham ao conjunto de dados original, enquanto o Discriminador é treinado para distinguir entre as amostras reais e as amostras geradas pelo Gerador. Esses dois componentes são treinados em conjunto de forma adversária, como que em uma competição em que o Gerador tenta enganar o Discriminador e o Discriminador busca cada vez mais distinguir as amostras reais das sintéticas (GOODFELLOW et al., 2014).

Contudo, enquanto o GAN tradicional tem como objetivo principal gerar amostras sintéticas que se assemelham ao conjunto de dados original, o GAIN é projetado especificamente para lidar com a imputação de dados ausentes. Sendo assim, o GAIN vai além, pois incorpora um processo de imputação para preencher os valores ausentes.

Durante o treinamento, o GAIN incrementa uma etapa adicional onde o Gerador aprende a imputar valores ausentes com base nas relações presentes nos dados observados. Isso significa que o GAIN não apenas gera amostras sintéticas realistas, mas também preenche lacunas nos dados ausentes, aproveitando as vantagens do GAN em termos da geração de dados de alta qualidade.

Uma das diferenças essenciais entre o GAIN e o GAN tradicional é a inclusão dessa etapa de imputação durante o treinamento adversário. Enquanto no algoritmo GAN clássico o objetivo é apenas gerar amostras sintéticas realistas, o GAIN enfatiza a imputação precisa de lacunas. Além disso, no GAIN, o Discriminador é treinado para distinguir entre os dados observados e os dados imputados, em vez de apenas diferenciar entre dados

reais e sintéticos. Essa competição entre o Gerador e o Discriminador é fundamental para garantir que as imputações sejam plausíveis e indistinguíveis dos dados reais.

O GAIN oferece uma abordagem para a imputação de dados ausentes, aproveitando as capacidades de geração de dados sintéticos de alta qualidade do GAN. Ao adicionar uma etapa de imputação durante o treinamento adversário, o GAIN permite que o Gerador aprenda a preencher os valores ausentes com base nas informações contidas nos dados observados, resultando em imputações mais realistas (YOON; JORDON; SCHAAR, 2018).

O GAIN combina a capacidade de modelagem dos dados com a habilidade de avaliar a qualidade das estimativas geradas. Este método é baseado em algoritmos de redes neurais que consiste em duas partes principais: a rede neural Geradora (Generator) e a rede neural Discriminadora (Discriminator). O Gerador é responsável por gerar valores substitutos para os dados ausentes, enquanto o Discriminador avalia a qualidade dessas substituições em relação aos dados reais, as duas partes são treinadas utilizando o treinamento adversário.

O treinamento adversário é uma competição entre o gerador e o discriminador, onde o gerador busca enganar o discriminador para que não seja capaz de distinguir corretamente entre os dados reais e os dados imputados. Por sua vez, o discriminador busca melhorar sua capacidade de discernimento e identificar com precisão os dados reais.

Durante o treinamento, os pesos do gerador e do discriminador são atualizados iterativamente usando técnicas de otimização, como por exemplo a descida do gradiente, para minimizar a diferença entre a distribuição dos dados completos reais e a distribuição dos dados imputados pelo gerador (YOON; JORDON; SCHAAR, 2018).

Ainda nas etapas de treinamento, o algoritmo GAIN opera em duas etapas principais: geração de substituições (*substitute generation*) e treinamento adversário (*adversarial training*). Na etapa de geração de substituições, o Gerador é responsável por produzir estimativas para os dados ausentes com base nas informações disponíveis nos dados observados. Essa etapa não requer um pré-treinamento.

Após a geração das substituições, inicia-se a etapa de treinamento adversário. Nessa etapa, o Gerador e o Discriminador são combinados e treinados em conjunto. O Gerador gera substituições para os dados ausentes, enquanto o Discriminador avalia a qualidade dessas substituições, tentando distinguir entre os dados reais e os dados imputados. Estes competem entre si, onde o Gerador busca gerar substituições cada vez mais realistas para enganar o Discriminador. Ao final do treinamento, o Gerador é capaz de gerar imputações para os dados ausentes que são estatisticamente plausíveis e preservam as características do conjunto de dados original (WANG et al., 2021).

Durante o treinamento adversário, o GAIN utiliza uma tabela de dicas (*Hints*), que indica a presença ou ausência dos dados em cada entrada, para orientar a imputação dos valores ausentes. Essa tabela fornece informações adicionais sobre a estrutura e os padrões

dos dados observados, ajudando o Gerador a gerar estimativas mais precisas (YOON; SULL, 2020).

Para melhor compreensão é importante detalhar o conceito e funcionalidade tanto do Gerador como do discriminador. O Gerador no contexto do GAIN refere-se a uma rede neural responsável por gerar substituições para os valores ausentes em um conjunto de dados. Assim trata-se de uma parte fundamental do algoritmo GAIN, que visa imputar dados ausentes de forma precisa e realista.

O Gerador caracteriza-se como uma rede neural que recebe como entrada uma matriz incompleta de dados, onde alguns valores estão ausentes, e produz como saída uma matriz completa com substituições para os valores ausentes. Essas substituições são geradas com base nas relações e padrões presentes nos dados observados.

O Gerador no contexto do GAIN trata-se de uma rede generativa. Esse tipo de rede refere-se a um tipo de modelo de aprendizado de máquina projetado para gerar novos exemplos de dados que se assemelham ao conjunto de dados original. Essas redes são capazes de aprender a distribuição de probabilidade dos dados observados e, a partir disso, gerar amostras sintéticas que se assemelham às amostras reais (ARJOVSKY; CHINTALA; BOTTOU, 2017).

Assim, no contexto do presente estudo, o Gerador é responsável por gerar substituições para os valores ausentes em um conjunto de dados. Ele aprende, portanto, a partir de dados observados para imputar valores ausentes de forma plausível, onde a rede generativa do GAIN é treinada em conjunto com o Discriminador, que avalia a qualidade das substituições geradas pelo Gerador. O Gerador por sua vez busca gerar substituições que sejam indistinguíveis dos dados reais, enganando o Discriminador.

As redes generativas são frequentemente implementadas utilizando arquiteturas de redes neurais, como redes neurais convolucionais (CNNs) ou redes neurais recorrentes (RNNs). Essas redes possuem camadas ocultas que transformam as entradas em uma representação latente, e uma camada de saída que gera as amostras sintéticas. Durante o treinamento, os pesos e parâmetros da rede são ajustados para otimizar a capacidade de geração de dados sintéticos realistas (GOODFELLOW, 2017).

Geralmente, o Gerador é projetado como uma rede neural *feedforward*, que consiste em várias camadas ocultas e uma camada de saída. Cada camada contém um conjunto de neurônios que executam operações matemáticas, como multiplicação de matrizes e ativações não lineares. A arquitetura específica do Gerador pode variar dependendo do problema e do domínio de dados em questão.

Durante o treinamento do GAIN, é importante destacar também que Gerador é atualizado usando técnicas de otimização, como descida de gradiente, para ajustar os pesos e os parâmetros da rede neural. O objetivo é minimizar a diferença entre as substituições

geradas pelo Gerador e os valores reais ausentes, de modo que as imputações sejam o mais próximo possível dos dados reais.

Já o Discriminador, desempenha um papel crítico na avaliação da qualidade das substituições geradas pelo Gerador. Em termos gerais, o Discriminador é uma rede neural projetada para distinguir entre amostras reais e amostras sintéticas. Ele recebe como entrada tanto dados reais observados quanto as substituições geradas pelo Gerador e gera uma saída que indica a probabilidade de a amostra ser real ou sintética. O objetivo do Discriminador é aprender a discernir entre os dados reais e os dados gerados pelo Gerador (SHAHBAZIAN; TRUBITSYNA, 2022).

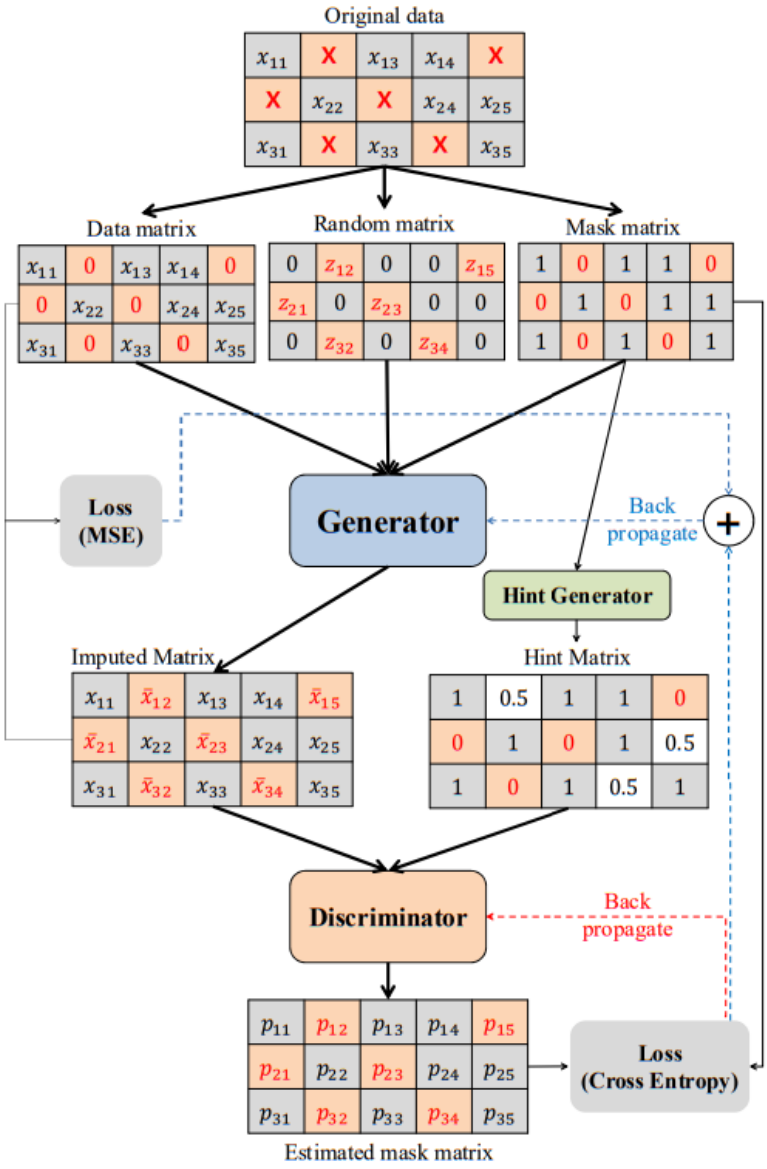
A arquitetura do discriminador pode incluir camadas densas (*fully connected*) ou convolucionais, seguidas por camadas de ativação e uma camada de saída que fornece a probabilidade de uma amostra ser real ou sintética (SHAHBAZIAN; TRUBITSYNA, 2022).

As redes discriminativas são, portanto, um tipo de arquitetura de rede neural que tem como objetivo aprender a distinguir diferentes classes de dados ou realizar tarefas de classificação. No contexto do GAIN, o Discriminador é uma rede discriminativa que desempenha um papel fundamental no treinamento do modelo.

Durante o treinamento, o Discriminador aprende a distinguir entre as duas categorias de amostras, ajustando seus pesos e parâmetros para maximizar sua capacidade de discriminação. À medida que o treinamento avança, espera-se que o Discriminador se torne cada vez mais habilidoso em identificar as substituições geradas pelo Gerador, diferenciando-as das amostras reais. Essa competição entre o Gerador e o Discriminador é o que impulsiona o aprimoramento do modelo GAN e, por extensão, do GAIN (YOON; SULL, 2020).

Assim, no contexto do GAIN, o Discriminador não apenas avalia a autenticidade das substituições geradas pelo Gerador, mas também fornece *feedbacks* para ajustar os pesos e hiperparâmetros do Gerador durante o treinamento adversário. Isso permite que o Gerador aprenda a imputar valores ausentes de forma mais precisa, aproveitando as informações e relações presentes nos dados observados. Na Figura 1 tem-se uma ilustração do funcionamento do GAIN (YOON; JORDON; SCHAAR, 2018).

Figura 1 – Arquitetura do GAIN proposta por Yoon, Jordon e Schaar (2018)



Fonte: Yoon, Jordon e Schaar (2018).

No topo da Figura 1 observa-se a matriz *Original Data* de dimensão $i \times j$, onde i representa as linhas da matriz e j as colunas. Nela tem-se os dados na forma como são captados, ou seja, em seu formato original, com a presença de lacunas no conjunto de dados. Estas estão representadas pelos espaços em vermelho da figura.

Logo abaixo tem-se três matrizes de mesma dimensão da matriz de dados originais. Estas foram nomeadas da seguinte forma: *data matrix, random matrix e mask matrix*. A matriz *data matrix* trás a representação da substituição das lacunas da matriz de dados original por zero.

Tem-se também a *Random matrix*, chamada de matriz de ruído, que representa uma matriz em que as lacunas são preenchidas com dados aleatórios, gerado a partir de

uma distribuição uniforme, e os demais campos da matriz original nesta terceira matriz são preenchidos com zeros. Já a *Mask matrix* trás uma máscara da matriz de dados original, onde os dados observados são rotulados como 1 e as lacunas são rotuladas com 0. As três últimas matrizes citadas (*Data matrix, Random matrix e Mask Matrix*), são passadas como *input* para a rede generativa.

$$G : \tilde{X} \times \{0,1\}^d \times [0,1]^d -> X \tag{2.5}$$

Na Equação 2.5, $\tilde{X}$ representa os valores imputados em cada posição com lacuna da *Data matrix*. Já o intervalo $\{0,1\}^d$ expressa os valores que podem ser assumidos dentro desse intervalo de 0 a 1 elevado ao tamanho $\boldsymbol{d}$ da dimensão da matriz. Enquanto que $[0,1]^d$ assume os valores binários que compõem a *Mask matrix* (POPOLIZIO et al., 2021).

A Equação 2.5 produz como saída o vetor de valores imputados, onde para obter o vetor de dados completo substitui-se os dados ausentes pelos valores imputados correspondentes, presentes em $\overline{X}$(*Imputed Matrix*).

Obtem-se, portanto, como saída a *Imputed Matrix*, onde os dados faltantes são imputados. Vale ressaltar que a matriz de imputações do Gerador é passada para uma função de perda (MSE), que ajudará o gerador por meio do ajustes dos pesos da rede neural a melhorar as imputações nas próxima vezes que ele for executado.

Em paralelo a geração da *Imputed Matrix* com base na *Mask matrix*, uma matriz de dicas é gerada: a *Hint Matrix* (Figura 1). Esta trás a informação de qual dos valores em cada posição da matriz é real, qual é falso e de qual valor não se tem informações. A ideia da tabela de dicas é melhorar as imputações realizadas pelo Gerador, pois leva o discriminador a fazer uma análise mais precisa da qualidade das imputações realizadas.

Baseado na *Imputed Matrix* e na *Hint Matrix*, o discriminador (*discriminator*), gera uma matriz de probabilidades de acerto do Gerador (*Estimated mask matrix*). Neste ponto, a *Estimated mask matrix* serve como entrada para uma outra função chamada *Cross Entropy*, responsável por especificar o quanto o Gerador acertou na imputação dos dados e quanto o Discriminar acertou na diferenciação dos dados reais e sintéticos criados pelo gerador. Após isso essa informação será enviada para o gerador e discriminador, por meio de uma retropropagação (*Back propagate*).

Portanto, a tarefa do Discriminador (D) é prever a máscara M do amostra completa. Então, a saída de D é um vetor cujos elementos são probabilidades; precisamente, o i-ésimo componente deste vetor é a probabilidade do dado ter sido observado (POPOLIZIO et al., 2021).

Sendo assim, o processo ilustrado na Figura 1 representa a disputa entre o gerador e o discriminador que é replicado por um número específico de vezes, as épocas. Desta

forma ao final de todas as épocas, o Gerador estará apto para realizar as imputações em um conjunto de dados de teste, por exemplo.

2.3 Métricas de avaliação

Nesta seção serão detalhadas as métricas utilizadas como parâmetros para medir o desempenho dos algoritmos avaliados neste estudo quanto a qualidade de suas imputações.

2.3.1 MAE - Mean Absolute Error

O MAE (*Mean Absolute Error*), ou Erro Médio Absoluto, é uma métrica comumente utilizada para avaliar o desempenho de modelos de aprendizado de máquina em problemas de regressão e é amplamente aplicada em diversas áreas, como economia, finanças, medicina e ciência dos dados. Ela mede a diferença média absoluta entre as previsões feitas pelo modelo e os valores reais dos dados (SHAHBAZIAN; TRUBITSYNA, 2022) e (LITTLE; RUBIN, 2019).

A Equação 2.6 mais adiante é formulada para calculo o MAE. Dado um conjunto de previsões $y_{\text{pred},i}$ e um conjunto de valores reais $y_{\text{true},i}$, o MAE é obtido calculando-se a média das diferenças absolutas entre cada previsão e seu valor real correspondente, onde n representa o número total de exemplos no conjunto de dado.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_{\text{pred},i} - y_{\text{true},i}| \tag{2.6}$$

Na Equação 2.6 tem-se a expressão para o cálculo do MAE, que apresenta o somatório da diferença, em valores absolutos, entre o valor previsto e o valor real da variável em análise, com os dados variando de 1 a n, dividindo esse valor pelo número total de exemplos do conjunto de dados.

O MAE é uma métrica menos sensível a valores atípicos (outliers) em comparação com outras métricas, como o erro médio quadrático (MSE). Enquanto o MSE penaliza de forma mais intensa grandes erros, o MAE trata todos os erros de forma igual, o que pode ser preferível em determinados contextos (LITTLE; RUBIN, 2019).

Na Equação 2.6, nota-se que o *Mean Absolute Error* não fornece informações sobre a direção dos erros, ou seja, torna-se difícil vizualizar se as previsões estão superestimando ou subestimando os valores reais. Dessa maneira, para conseguir abordar essa limitação, outras métricas são adotadas, como por exemplo, o RMSE (*Root Mean Squared Error*) que será descrito na subseção 2.3.2.

2.3.2 RMSE - Root Mean Squared Error

Além do MAE, uma métrica igualmente utilizada para avaliar o desempenho de modelos de regressão é o RMSE. Este, assim como o MAE é amplamente adotado na análise de modelos de previsão como métrica estatística padrão para medir o desempenho desse modelo em estudos nas áreas de meteorologia, qualidade do ar e pesquisas climáticas, por exemplo (CHAI; DRAXLER, 2014) e (YOON; SULL, 2020). Estes últimos autores oferecem uma perspectiva ligeiramente diferente sobre a qualidade do modelo.

Ao passo que o MAE confere o mesmo peso a todos os erros, o RMSE insere sobre a variância penalizando-a, tendo em vista que dá mais peso a erros com valores absolutos maiores, pois considera além da magnitude, a direção dos erros, proporcionando uma maior penalização para discrepâncias significativas entre as previsões e valores reais (CHAI; DRAXLER, 2014).

Alguns autores na literatura trazem que MAE e RMSE são métricas parecidas dado a sua estrutura matemática, onde apenas ao RMSE a natureza quadrática é acrescida. Contudo estas métricas não são equivalentes, conforme demonstrado por (WILLMOTT; MATSUURA, 2005). Estes autores trazem que o RMSE não é equivalente ao MAE e que não é tão simples e possível derivar o valor do MAE do RMSE e vice-versa.

Para calculo do RMSE utiliza-se a fórmula descrita na Equação 2.7.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{\text{pred},i} - y_{\text{true},i})^2} \tag{2.7}$$

A Equação 2.7 envolve a diferença ao quadrado entre as previsões $y_{\text{pred},i}$ e os valores reais $y_{\text{true},i}$ para cada exemplo.

A diferença de formulação matemática no cálculo do RMSE, o torna mais sensível a erros grandes do que o MAE. Portanto, em casos em que o modelo esteja sujeito a erros significativos, o RMSE destacará essas discrepâncias de forma mais evidente do que o MAE. Por fim, o RMSE permite avaliar se as previsões tendem a superestimar ou subestimar os valores reais. Assim, quanto mais próximo o RMSE for de zero, o modelo pode ser considerado com um melhor desempenho na tarefa de previsão.

Não há um consenso na literatura sobre qual melhor métrica para mensuração de erros em modelos. O uso dependerá das necessidades do problema. Autores como (CHAI; DRAXLER, 2014) destacam que por vezes, mais de uma métrica são necessárias para fornecer uma visão mais completa da distribuição de erros. Em nosso estudo selecionamos as duas mmétricas apresentadas nesta seção para avaliar o desempenho dos modelos em estudo.

3 Trabalhos Relacionados

Diversos autores concentram seus esforços na implantação de técnicas de imputação de dados como solução para preenchimento de lacunas em conjunto de dados nos mais diversos contextos de aplicação. Para Oliveira et al. (2021), o uso do método de imputação de dados, KNN é tratado como a solução ao problema de lacunas de dados para assim conseguir realizar a aplicação de métodos de predição de valores futuros de séries temporais.

Em seu estudo, os autores utilizam o KNN para encontrar registros completos mais próximos do registro ausente, substituindo as lacunas pelo valor médio encontrado baseado nos vizinhos mais próximos da respectiva variável para cada valor ausente encontrado. O método KNN também foi utilizado no presente estudo para apoiar na Avaliação de Redes Adversárias Generativas, que também constitui-se de um método para imputação múltipla de séries temporais, realizando um comparativo em relação ao desempenho destas frente à algoritmos mais tradicionais como o KNN que comumente é utilizado na literatura, como observa-se.

Outros autores como Carvalho et al. (2017) trouxeram um comparativo entre modelos de imputação múltipla e modelos geoestatísticos de *Kriging* e *Co-Kriging* para estimar dados diários de precipitação e preenchimento de falhas. Em seu estudo os autores ressaltaram que a aplicação MICE produziu melhores estimativas dos valores diários de precipitação do que os modelos geoestatísticos, sendo indicado como alternativa para preencher lacunas em séries temporais de precipitação.

Assim como o autor supracitado, nosso estudo também trouxe essa linha comparativa, contudo ao invés da adoção de modelos geoestatísticos, utilizou-se redes adversárias generativas, como o GAIN para imputação e com um olhar clínico buscou-se, assim como Carvalho et al. (2017), observar o desempenho deste frente ao MICE, bem como com relação ao KNN como já fora supracitado.

Em seu estudo Carvalho et al. (2017) ressaltaram ainda a importância da imputação de dados meteorológicos e seu impacto na tomada de decisões relacionadas ao monitoramento do transporte marítimo, aviação civil e militar, agricultura, saúde, energia, prevenção de desastres naturais entre outras aplicações, onde falhas em séries históricas seriam catastróficas na previsão das condições ideias de controle destes cenários. Estas múltiplas aplicações trazidas pelos autores demonstram a relevância do presente estudo para investigação e análise de métodos de imputação de maneira a identificar possíveis pontos de melhoria na qualidade das imputações com uma abrangência de aplicação para diferentes cenários relevantes do cotidiano da sociedade tendo influência e impacto direto na mesma.

Alguns autores destacam os métodos de imputação de dados por categorias. Tendo-se aqueles mais tradicionais, como os que adotam deletar observações que estão parcialmente ausentes (método de exclusão) ou os que utilizam-se do método de imputação que imputa os valores ausentes com valores razoáveis como é o caso do KNN, imputação baseada em fatoração de matrizes e imputação baseada em Redes Neurais Recorrentes (RNN) ou uma abordagem mais sofisticada como é o observado nos métodos baseados em GAN, no caso o GAIN (LUO et al., 2019). Observar métodos que ofereçam uma visão menos simplista, do que simplesmente eliminar e perder parte dos dados devido a presença de lacunas é uma alternativa que permite trazer análises mais completas e precisas sobre um fenômeno em determinada população.

Assim como no nosso estudo, o método proposto por Luo et al. (2019) busca aprender automaticamente as representações internas da série temporal, contudo com a diferença que este tenta aprimorar a reconstrução dos dados utilizando-se de uma estratégia de compressão e reconstrução para evitar o que o autor chama de estágio de otimização do "ruído". Os autores propõem, portanto, a utilização do modelo de imputação GAIN, para imputação de séries temporais dividido em dois estágios: treinamento de um modelo GAN onde para cada amostra, ele tenta otimizar o vetor de entrada de "ruído" e encontrar o melhor vetor de entrada correspondente do gerador para se ter uma amostra mais semelhante a original.

Luo et al. (2019), também cita alguns métodos para preenchimento de lacunas e os caracteriza como razoáveis para esse tipo de problema de ausência de dados: *K-Nearest Neighbor* (KNN), imputação baseada em fatoração de matriz e imputação baseada em redes neurais recorrentes (RNN) (HUDAK et al., 2008; MORUP et al., 2010; PEURIFOY, 2018; CHE et al., 2018; CAO et al., 2018).

Com base no que já fora dito, nota-se que o método GAIN aparece na literatura como solução para lidar com o problema de dados faltantes em conjuntos de dados meteorológicos. Um estudo recente realizado por Popolizio et al. (2021) explorou o desempenho do GAIN nesse contexto, comparando-o com outro método conhecido na literatura, o ARMA (*Autoregressive Moving Average*), utilizando o erro quadrático médio (RMSE) como métrica de avaliação.

No trabalho de Popolizio et al. (2021), foram realizados testes numéricos para avaliar a eficácia do método GAIN em imputar dados faltantes em conjuntos de dados meteorológicos. Os resultados obtidos demonstraram que o GAIN superou o método ARMA em termos de desempenho, reduzindo o RMSE e melhorando a qualidade das imputações. Eles observaram um comportamento muito semelhante entre os valores obtidos na métrica RMSE para o GAIN, em detrimento do ARMA. Esse distanciamento entre os valores da métrica RMSE para o conjunto de treinamento e o de teste mostra que GAIN não superajusta os dados e é capaz de preservar o seu valor, bem como características

específicas do conjunto de dados, demonstrando também que as redes neurais aprenderam o modelo subjacente aos dados da série temporal e que eles são capazes de generalizar seu comportamento no caso do GAIN.

Os resultados observados por Popolizio et al. (2021) fornecem evidências de que o método GAIN é uma abordagem interessante para imputação de dados faltantes em conjuntos de dados meteorológicos, que são alvos de nosso estudo, quando nos propomos a analisar dados de temperatura do estado de Pernambuco.

A utilização de testes numéricos e a comparação com um método conhecido na literatura fortalecem a validade do uso de métodos como o GAIN para a imputação de dados e despertam a necessidade de maior investigação e avaliação desse método tendo em vista sua ampla aplicabilidade e desempenho.

Os autores Wu et al. (2022) propuseram uma estrutura aprimorada para o método de imputação multivariada utilizando as equações encadeadas (MICE) para lidar com lacunas de dados. O estudo teve como objetivo comparar diferentes algoritmos, incluindo MICE e GAIN, para imputação de dados em um desafio específico de preenchimento de grandes lacunas de dados, onde os únicos dados relacionados são provenientes da mesma estação de sensor.

No trabalho de Wu et al. (2022), o método MICE demonstrou maior acurácia em comparação com o GAIN para a inserção de lacunas de dados. Esse resultado sinaliza que o MICE pode ser mais eficaz na imputação de grandes lacunas de dados em cenários onde apenas dados provenientes da mesma estação de sensor estejam disponíveis. Essa comparação entre MICE e GAIN fornece informações interessantes para a escolha do método de imputação em desafios específicos de preenchimento de grandes lacunas de dados, demonstrando que o contexto e características específicas de cada conjunto de dados pode influenciar diretamente nos resultados de desempenho obtidos através da utilização de diferentes métodos de imputação.

Os resultados dos estudos demonstram que cada método possui vantagens e limitações, e a escolha do método adequado depende do contexto e das características específicas dos dados. O GAIN e o MICE são métodos mais sofisticados, com maior variância, com potencial para lidar com lacunas em séries temporais complexas, enquanto o KNN é uma abordagem mais simples e direta, adequada para conjuntos de dados com estrutura local bem definida.

Os estudos revisados, descritos na Tabela 1, fornecem embasamento teórico e posicionam o presente estudo com relação ao uso de métodos para a imputação de dados em séries temporais multivariadas em diversos contextos. Na próxima seção apresentaremos os métodos a serem avaliados neste estudo bem como as condições adotadas no ambiente de experimentação, dados utilizados e tratamento, bem como métricas de avaliação.

Tabela 1 – Trabalhos relacionados e campo de aplicação

Trabalho	**Método de imputação**	**Campo de aplicação**
(CARVALHO et al., 2017)	*Multiple Imputation by* textitChained Equations (MICE).	Meteorologia
(LUO et al., 2019)	*k-Nearest Neighbors* (kNN), *Matrix Factorization* (MF), GRUD, GAN-2-stage, BRITS e GAIN.	Medicina e Meteorologia
(OLIVEIRA et al., 2021)	*k-Nearest Neighbors* (kNN).	Meteorologia
(POPOLIZIO et al., 2021)	*Auto Regressive Moving Average*(ARMA) e GAIN.	Meteorologia
(WU et al., 2022)	*k-Nearest Neighbors* (kNN), Linear, MICE-SVR, MICE- *Decision Tree*, MICE-*Random Forest*, GAIN, MRNN e LSTM-RNN.	Meteorologia

4 Materiais e métodos

Neste capítulo serão apresentados os materiais utilizados na aquisição e processamento dos dados, bem como os métodos e métricas de avaliação utilizados nas experiências e estudos dessa dissertação.

4.1 Dados

Os dados disponibilizados para a realização dos experimentos foram os dados horários de temperatura máxima de 20 estações meteorológicas do estado de Pernambuco correspondentes ao período de setembro de 2019 a abril de 2022 disponibilizados pela Agência Pernambucana de Águas e Clima - APAC[1] que foram armazenados em formato bruto em um banco de dados e disponibilizados via API (*Application Programming Interface*) pela ferramenta Thingsboard[2], que está implantada em um servidor disponibilizado pela Universidade Federal Rural de Pernambuco - UFRPE.

Para o uso dos dados nos experimentos foi realizado um pré-processamento, com o objetivo de encontrar as estações com um mesmo período de dados completos. Desta forma, das 20 estações iniciais, foram encontradas 11 estações no período de 19/11/2021 01:00:00 a 02/01/2022 23:59:59 que estavam com o período de dados completo, ou seja, sem lacunas. Essa limpeza foi necessária para a inserção de taxas (variando de 5 a 50%) de lacunas de dados de forma que fosse possível ter maior controle do experimento e assim realizar a análise por meio das métricas MAE e RMSE (seção 2.3) .

O pré-processamento resultou em um conjunto com 1079 dados horários para cada estação, onde cada linha do conjunto correspondeu a um registro. Esses dados estão representados no formato de séries temporais, de forma que cada série corresponde à variável de temperatura máxima e possui registros em intervalos horários.

Cada série está associada também a uma estação meteorológica que foi analisada neste estudo. Além disso, os dados foram divididos em dados de treino, validação e teste, sendo 80% para treino, 10% para validação e 10% para teste, resultando ao final em conjuntos de dados para cada variação de taxa de lacunas que será melhor explanado na seção 4.4.

[1] https://www.apac.pe.gov.br/
[2] https://thingsboard.io/

4.2 Ambiente de experimentação

Para a realização dos experimentos utilizou-se o *Google Colab*[4], que é uma plataforma baseada em nuvem que permite criar, executar e colaborar em ambientes interativos de computação que permitem combinar código, texto formatado, equações, visualizações e outros elementos em um único documento. Essa plataforma é amplamente utilizada por cientistas de dados, pesquisadores, desenvolvedores e educadores para realizar análises de dados, explorar ideias, criar demonstrações interativas e compartilhar conhecimentos.

Dentro do contexto das bibliotecas do *Python*, utilizou-se as bibliotecas do *TensorFlow* (ABADI et al., 2015) e *keras* (CHOLLET, 2015) para o GAIN, já para o *KNN Imputer* e MICE, fez-se uso da biblioteca *scikit-learn* (PEDREGOSA et al., 2011).

Para o armazenamento e disponibilização via API e visualização dos dados brutos coletados da APAC utilizou a ferramenta *Thingsboard*, a qual possui um banco de dados *PostgreSQL*[6] integrado. *Thingboard* é uma plataforma de código aberto para a Internet das Coisas (IoT) que permite a coleta, processamento, visualização e gerenciamento de dados de dispositivos conectados. É uma ferramenta poderosa para construir aplicações IoT escaláveis e personalizadas, e é projetada para facilitar o desenvolvimento rápido e eficiente de soluções IoT.

4.3 Algoritmos e configurações

Para realização das imputações das lacunas de dados de temperatura máxima e avaliação das Redes Adversárias, para a imputação de dados na fase de experimentos deste trabalho fez-se uso dos algoritmos: KNN (*KNN Imputer)*, MICE e GAIN para imputação de dados.

A escolha dos métodos foi determinada de acordo com as características dos dados e os objetivos da pesquisa. Essa seleção foi baseada na capacidade de cada método em lidar com as complexidades dos dados de séries temporais meteorológicas, bem como na abordagem específica de cada algoritmo para enfrentar os desafios da imputação.

O KNN é conhecido por sua eficácia em encontrar padrões locais em conjuntos de dados de baixa dimensionalidade, enquanto o MICE lida bem com relações complexas entre variáveis. Já o GAIN, embora seja um método mais recente, chamou a atenção para uso neste trabalho por sua abordagem baseada em redes adversárias generativas, que pode capturar características latentes nos dados. Além disso, o KNN e MICE, são algoritmos comumente utilizados em trabalhos de imputação de dados e o GAIN é um algoritmo que no melhor do nosso conhecimento não havia sido avaliado em dados meteorológicos de

[4] https://colab.research.google.com/
[6] https://www.postgresql.org/

temperatura máxima da APAC no estado de Pernambuco (JING et al., 2022; YANG et al., 2020).

O **KNN Imputer** é um algoritmo de imputação de dados que se baseia em um funcionamento simples e intuitivo. Ele utiliza as informações dos vizinhos mais próximos para estimar valores ausentes em cada variável da série temporal. Ao identificar os k exemplos mais próximos com valores conhecidos, o KNN Imputer calcula uma estimativa para os valores faltantes, levando em conta a proximidade em relação aos vizinhos. Essa abordagem de imputação torna esse método uma opção popular para tratar dados faltantes em séries temporais, permitindo uma imputação eficiente e precisa com base na estrutura temporal dos dados, conforme está descrito na subseção 2.2.1.

O uso do algoritmo KNN Imputer envolveu a definição de alguns parâmetros para o preenchimento das lacunas em um conjunto de dados. Esses parâmetros são fundamentais para determinar o comportamento do algoritmo e a qualidade das imputações realizadas. Os principais parâmetros utilizados foram:

Métrica de distância: a métrica de distância é uma medida que determina a proximidade entre os exemplos no conjunto de dados. Apesar de ter sido utilizada a métrica de distância euclidiana, mas também é possível escolher outras métricas, como a distância de Manhattan ou a distância de Minkowski, dependendo do contexto e da natureza dos dados.

O valor de **k** representa o número de vizinhos mais próximos a serem considerados durante o processo de imputação. Um valor muito baixo pode levar a uma imputação ruidosa, enquanto um valor muito alto pode resultar em uma imputação tendenciosa. Utilizou-se, portanto, o valor de k igual a 5, como melhor valor encontrado para o k tomando como base análises realizadas com os dados disponíveis nesse trabalho através de busca em grade, além de observar os valores de k comumente encontrados na literatura (ADDI et al., 2022; LUO et al., 2019).

O **MICE** é outro algoritmo utilizado na imputação de dados (subseção 2.2.2), pois trata-se de um método estatístico utilizado para tratar lacunas em análises estatísticas. Ele é amplamente utilizado para lidar com dados incompletos, onde há valores ausentes em algumas observações. O método MICE envolveu a imputação (preenchimento) dessas lacunas de dados usando uma abordagem iterativa, onde várias equações de regressão foram usadas para prever os valores ausentes com base em outros dados observados.

A aplicação do método MICE também envolve a definição de parâmetros para o preenchimento de lacunas em um conjunto de dados. Esses parâmetros desempenham um papel fundamental ao determinar o comportamento do algoritmo e a qualidade das imputações realizadas. Os parâmetros utilizados neste trabalho para MICE tomam como base análises realizadas com os dados disponíveis nesse trabalho através de busca em

grade, bem como a exploração dos parâmetros frequentemente utilizados em trabalhos semelhantes. Os principais parâmetros e seus respectivos valores utilizados no MICE estão descritos na Tabela 2.

Tabela 2 – Parâmetros Utilizados no MICE

Parâmetro	**Valor**
Número de iterações	10
Método de imputação inicial	Média Aritmética
Método de imputação	Regressão linear
Critério de convergência	0.001

Na Tabela 2 o **número de iterações** determina quantas vezes o processo de imputação foi repetido. Nesse aspecto, um número suficiente de iterações foi definido de maneira a garantir a convergência das imputações.

No parâmetro **método de imputação inicial**, antes das iterações, as lacunas são preenchidas inicialmente usando a média arimética dos valores presentes no conjunto original. Em seguida, o parâmetro **Método de imputação** utilizado pelo MICE, em nosso estudo, foi a Regressão linear, dado a sua melhor aderência aos dados. O parâmetro **critérios de convergência**, 0,001, foi utilizado para definição de pontos de parada que indicassem quando as imputações atingiriam a convergência, ou seja, quando os valores imputados estariam estáveis o suficiente.

Além do KNN e MICE, foi avaliado ainda o algoritmo GAIN, que é um método de imputação de dados que utiliza a abordagem de redes adversárias para tratar lacunas em conjuntos de dados. Inspirado pela estrutura das redes generativas adversárias (GANs), o GAIN propõe um modelo composto por um gerador e um discriminador que trabalham juntos para realizar as imputações.

Como está detalhado na subseção 2.2.3, esse método é uma abordagem complexa que envolve redes neurais e aprendizado de máquina. Portanto, a seleção dos parâmetros pode variar de acordo com a implementação específica e o conjunto de dados utilizado. No entanto, nota-se alguns parâmetros importantes e comumente utilizados no GAIN, que são: número de camadas ocultas, número de iterações de treinamento, épocas, taxa de aprendizado (learning rate), número de neurônios, taxa de dicas (hints), tamanho do lote, funções de ativação e otimizador. Vale ressaltar que os hiperparâmetros do GAIN descritos a seguir, foram definidos com base em busca aleatória (*Random Search*), que terá uma abordagem mais detalhada na seção 4.4.

O **número de camadas ocultas** determina quantas camadas ocultas as redes neurais do gerador e do discriminador terão. Uma estrutura mais profunda pode permitir a aprendizagem de representações mais complexas.

O **Número de iterações de treinamento** define quantas iterações de treinamento serão realizadas. É importante ressaltar que, quanto mais iterações são realizadas mais consegue-se melhorar a qualidade das imputações, contudo isso também podem levar a um treinamento mais demorado.

As **épocas (*epochs*)** se referem a uma iteração completa de treinamento através de todo o conjunto de dados disponível. Uma época consiste em alimentar todos os exemplos de treinamento (amostras) no modelo, calcular as perdas, atualizar os pesos das redes e repetir esse processo para um número definido de vezes.

A **Taxa de aprendizado (*learning rate*)** controla o tamanho dos passos que o otimizador dá para ajustar os pesos das redes neurais durante o treinamento. Uma taxa de aprendizado alta pode levar a oscilações ou divergência, enquanto uma taxa muito baixa pode resultar em um treinamento lento.

O **Número de neurônios nas camadas ocultas** que foi utilizado define o número de neurônios em cada camada oculta das redes neurais do gerador e do discriminador. Esse número pode afetar a capacidade do modelo de capturar padrões complexos.

A **Taxa de dicas (*Hint*)** se refere à proporção de valores conhecidos ou "dicas" que foram fornecidas para orientar ou guiar o processo de imputação de dados faltantes no modelo. Em contextos de imputação de dados, especialmente em técnicas como o GAIN, a taxa de dicas representou a quantidade de informações disponíveis para o modelo durante a imputação.

O **Tamanho do lote (*Batch Size*)**: indica quantos exemplos de treinamento foram processados em cada iteração durante o treinamento. Um tamanho de lote maior pode ser responsável por acelerar o treinamento, mas também requer mais memória.

As **Funções de ativação**, ReLU (*Rectified Linear Unit*), Softplus, Softsign, Tanh, Selu, Elu, Exponential, LeakyReLU, Swich, Softmax, Gelu, Softign, Hard_sigmoid foram utilizadas nas camadas das redes neurais para introduzir não linearidade de maneira aleatória definida pelo *Random Search*.

Por fim, tem-se, **Otimizador (*Optimizer*)** que descreve o algoritmo responsável por ajustar os parâmetros do modelo de forma a minimizar ou maximizar uma função de perda, também conhecida como função objetivo. O objetivo principal do otimizador foi guiar o modelo em direção a um conjunto de parâmetros que levasse a um desempenho ideal na tarefa em questão. Este e os demais hiperparâmetros foram definidos através de busca aleatória, como já fora supracitado.

Outro questão fundamental, além dos hiperparâmetros, sobre GAIN no contexto de Redes Neurais é a função de perda (*Loss Function*), que é uma medida que quantifica o quão bem o modelo está performando em relação aos dados de treinamento e aos valores reais associados a esses dados. Ela desempenha um papel fundamental no treinamento de

redes neurais e em algoritmos de aprendizado de máquina em geral.

A principal função da *loss* (como é comumente chamada) é avaliar o quão distantes as previsões do modelo estão dos valores reais. A partir dessa medida de distância, o algoritmo de otimização ajusta os parâmetros da rede neural para minimizar a *loss* durante o processo de treinamento. Em outras palavras, a *loss* guia o aprendizado do modelo, orientando-o a encontrar os melhores parâmetros que resultem em previsões mais precisas (KINGMA; BA, 2014).

Tipicamente, as redes neurais têm como objetivo minimizar a *loss* durante o treinamento. No entanto, o tipo de *loss* usado pode variar de acordo com o tipo de problema. Para problemas de regressão, uma escolha comum é a *loss* quadrática, também conhecida como "*Mean Squared Error*" (*MSE*), que mede a média dos erros quadrados entre as previsões e os valores reais. Para problemas de classificação, a loss mais comum é a "*Cross-Entropy*", que mede a divergência entre as distribuições de probabilidades das previsões e dos rótulos reais.

Para este estudo, focando especificamente no método GAIN, que é constituído por duas Redes Neurais Adversárias (uma Geradora, conhecida como "Generator", e outra Discriminadora, chamada de "Discriminator"), foram adotadas funções de perda específicas para guiar o treinamento. Na rede Geradora, a função de perda aplicada é a quadrática (*MSE*), que calcula a média dos erros quadrados entre as previsões geradas e os valores reais.

Por outro lado, na rede Discriminadora, a função de perda escolhida é a *Cross-Entropy*, esta última mede a diferença entre as distribuições de probabilidades das previsões do modelo e os rótulos reais dos dados, a partir da matriz de máscara. As funções de perda foram selecionadas para otimizar o processo de treinamento das redes adversárias e melhorar a qualidade das imputações de dados faltantes (YOON; JORDON; SCHAAR, 2018). Na seção 4.4, tem-se detalhado o uso do método *Random Search* adotado com o objetivo de encontrar melhores hiperparâmetros para o GAIN dado a sua complexidade.

4.4 Random Search

O *Random Search* é um método de otimização utilizado em diversas áreas, como aprendizado de máquina, inteligência artificial e engenharia. Seu objetivo é encontrar os melhores parâmetros ou configurações para um determinado algoritmo ou modelo, maximizando ou minimizando uma métrica específica.

Diferentemente da busca em grade (*Grid Search*) que explora todos os pontos de uma grade fixa de valores, o *Random Search* seleciona aleatoriamente combinações de hiperparâmetros a serem avaliadas. Essa abordagem foi introduzida por James Bergstra

e Yoshua Bengio em seu trabalho *"Random Search for Hyper-Parameter Optimization"*, no qual eles mostraram que, mesmo com um número limitado de avaliações, o *Random Search* frequentemente supera a busca em grade em termos de eficiência e resultados (BERGSTRA; BENGIO, 2012).

Outra pesquisa que expandiu o entendimento sobre a eficácia do *Random Search* é o trabalho de James Bergstra, que explorou abordagens mais sofisticadas, como a adaptação de distribuições das quais os hiperparâmetros são amostrados aleatoriamente.

Neste trabalho, tomou-se para o *Random Search*, como espaço de busca, os valores apresentados na Tabela 3, disponíveis na biblioteca do *TensorFlow* (ABADI et al., 2015), *keras* (CHOLLET, 2015), nos materiais suplementares[7] do trabalho de (YOON; JORDON; SCHAAR, 2018; MENG et al., 2023), com exceção do número de camadas ocultas, onde neste trabalho adotou-se um total de 3 camadas. Para a arquitetura que recebe os valores otimizados pelo *Random Search* mais a adição de mais uma camada oculta nomeou-se de rede GAIN aprimorada.

Tabela 3 – Valores e distribuições dos Hiperparâmetros utilizados pelo *Random Search* para parametrizar o GAIN

Hiperparâmetro	**Valor**	**Distribuição**
Número de camadas ocultas em cada rede	3	-
Número de iterações de treinamento	30	-
Taxa de dicas (*Hint*)	[0 - 1]	Uniforme Contínua
Número de épocas (*Epoch*)	[10 - 100]	Uniforme Discreta
Taxa de aprendizado (*Learning Rate*)	[1×10^{-5} - 1×10^{-1}]	Loguniforme
Número de neurônios nas camadas ocultas	[50 - 400]	Uniforme Discreta
Tamanho do lote (*Batch Size*)	[32, 64, 128]	Uniforme Discreta
Funções de ativação (*Activation Function*)	[*ReLu, Softplus, Softsign, Tanh, Selu, Elu, Exponential, LeakyReLU, Swish, Softmax, Gelu, Softsign, Hard_sigmoid*]	Uniforme Discreta
Otimizador (*Optimizer*)	[*SGD, Adam, RMSprop, Adadelta, Adagrad, Adamax, Nadam, Ftrl*]	Uniforme Discreta

A Tabela 3 mostra os hiperparâmetros citados na seção 4.3, utilizados pelo *Random Search*, visando encontrar os melhores hiperparâmetros para otimizar o GAIN. Os

[7] https://proceedings.mlr.press/v80/yoon18a/yoon18a-supp.pdf

hiperparâmetros, **número de camadas ocultas e números de iterações de treinamento**, estão com valores absolutos. Para o **número de camadas ocultas** o valor (3). O hiperparâmetro **número de iterações de treinamento** foi definido com base nos recursos de infraestrutura disponíveis.

Ainda na Tabela 3 tem-se expresso o intervalo referente a taxa de dicas (*Hint*) que varia de 0 a 1, o número de épocas variando de 10 a 100 e o número de neurônios nas camadas ocultas que variando de 50 a 400. Esses intervalos são escolhidos de forma equiprovável dentre os valores listados na Tabela 3, sendo valores contínuos para a taxa de dicas e discreto para o número de épocas e de neurônios nas camadas ocultas.

Os hiperparâmetros tamanho do lote (*Batch Size*), funções de ativação (*Activation Function*) e otimizador são escolhidos de forma equiprovável dentre os valores listados na Tabela 3, com a presença de variáveis categóricas, conforme explícitas na Tabela 3.

O hiperparâmetro taxa de aprendizado, por sua vez, assume valores no intervalo 1×10^{-5} a 1×10^{-1}, onde esse intervalo segue uma distribuição *loguniforme* ou recíproca que abarca variáveis aleatórias contínuas, ou seja, trata-se de uma distribuição de probabilidade contínua em que o logaritmo da variável aleatória é uniformemente distribuído.

O processo para a busca dos melhores hiperparâmetros do GAIN foi aplicado na etapa de treino e validação deste estudo. Após essas duas últimas etapas do experimento os valores obtidos foram utilizados na fase de teste. Conforme dito na seção 4.1 o conjunto de dados real foi particionado em 3 partes: treino (80%), validação (10%) e teste (10%).

No conjunto de treino e de validação, foram inseridos 5% de lacunas, tendo em vista a premissa que um conjunto treinado com um menor percentual de lacunas tem um maior embasamento das características do conjunto de dados real. No conjunto de validação manteve-se o mesmo percentual de lacunas, variando apenas o conteúdo dos dados, ou seja, no conjunto de validação, o algoritmo já treinado com um mesmo percentual de lacunas agora seria validado para um conjunto de dados distinto, completamente novo.

No contexto dos conjuntos de treinamento e validação, empregamos uma variação da técnica de validação cruzada *'K-Fold'*. A abordagem utilizada é especialmente adaptada para lidar com dados de séries temporais e compartilha o objetivo fundamental de aumentar a confiabilidade do modelo de aprendizado de máquina. A ideia básica por trás da técnica de validação *'K-Fold'* é dividir os dados de treinamento e validação em vários conjuntos de mesmo tamanho (no caso K grupos) de forma independente e identicamente distribuída, e repetir K processos de treino e validação com estes grupos, usando K-1 grupos para treino e 1 grupo para validação.

Devido à natureza dos dados de séries temporais, nos quais os valores dependem dos valores anteriores de maneira sequencial, a aplicação direta da validação cruzada *KFold* tradicional não é apropriada, pois a série deixaria de ter a sua forma ordenada

(HASANOV; WOLTER; GLENDE, 2022). Portanto, utilizou-se nesse trabalho a técnica chamada de estratégia de divisão de séries temporais (essa abordagem também é conhecida como: janela deslizante, simulação deslizante, e até mesmo origem de previsão deslizante). Esta técnica é utilizada com dados divididos em K subconjuntos de mesmo tamanho (em nosso caso K=10), porém preservando a temporalidade dos dados. Inicialmente, o modelo é treinado com o primeiro conjunto e validado com o segundo conjunto, mas, a cada época (*Epoch*), o conjunto de treinamento é expandido com o conjunto de validação da rodada anterior (HASANOV; WOLTER; GLENDE, 2022).

Para o *Random Search*, nessas duas primeiras etapas do experimento foram executadas 30 iterações, onde para cada iteração em cada um dos conjuntos (treino e validação), a busca aleatória determinava o número de épocas a ser adotada na iteração, podendo variar de 10 a 100 épocas, conforme Tabela 3.

Cada iteração do *Random Search*, na etapa de validação, para cada época, retorna configurações de possíveis hiperparâmetros. O critério para a escolha do melhor conjunto de hiperparâmetros a serem utilizados na fase de teste, foi o valor médio das perdas do gerador (GMeanLoss). Maiores detalhes da fase de validação do GAIN serão explanados na seção 5.1.

Obtidos os melhores hiperparâmetros, seguiu-se para a fase de teste para se tentar avaliar a robustez do algoritmo ao treinamento com lacunas (seção 5.2. Nessa fase, foram inseridos diferentes percentuais de lacunas 5, 10, 20, 30, 40 e 50%, a fim de avaliar o desempenho do algoritmo frente a diferentes cenários. Para realizar a avaliação do desempenho dos algoritmos em apreço neste estudo, as métricas MAE e a RMSE foram utilizadas.

4.5 Métricas de Avaliação

As métricas de avaliação foram utilizadas para medir o desempenho e a eficácia de modelos, sistemas ou algoritmos em diferentes contextos. O uso de métricas de avaliação proporciona uma avaliação mais objetiva e quantitativa do desempenho do modelo ou sistema. Evitando assim decisões intuídas ou opiniões subjetivas, tornando o processo mais confiável e replicável.

Assim torna-se mais fácil quantificar o sucesso do modelo em atingir seus objetivos e identificar pontos de melhoria e refinamentos. Neste trabalho utilizou-se de métricas conhecidas na literatura (seção 2.3) como a MAE e RMSE para medir a qualidade da imputação dos dados.

A imputação de dados frequentemente envolve a avaliação do desempenho dos métodos utilizados, e é comum encontrar nas pesquisas as métricas de desempenho

MAE (Erro Absoluto Médio) e RMSE (Erro Quadrático Médio). Essas métricas são amplamente empregadas para medir a qualidade das imputações realizadas nos dados ausentes (FLORES; TITO; CENTTY, 2020; JING et al., 2022; YANG et al., 2020; MIR et al., 2022). Desta forma, a avaliação do desempenho dos algoritmos KNN *Imputer*, MICE e GAIN foi conduzida utilizando as métricas MAE e RMSE, como mencionado anteriormente.

5 Resultados e discussão

Neste capítulo serão apresentados os resultados obtidos utilizando os métodos apresentados no Capítulo 4. Será comentado sobre a validação e robutez do GAIN e por fim será feita a avaliação dos algoritmos clássicos, KNN e MICE, comparando-os com o método GAIN.

5.1 Validação dos hiperparâmetros do GAIN

Conforme detalhado na seção 4.4, o *Random Search* foi utilizado para busca dos melhores hiperparâmetros tanto na fase de treinamento quanto na validação. A validação foi feita utilizando-se dos mesmos hiperparâmetros obtidos na fase de treino pelo *Random Search* para um conjunto de dados diferente do que o algoritmo estava habituado (conjunto de validação).

O retorno obtido através do *Random Search* na etapa de validação, encontra-se na Tabela 4. O critério de escolha dos melhores hiperparâmetros foi aquele conjunto de hiperparâmetros que apresentasse o menor valor de *GMeanLoss* durante o experimento (0,001788). Um menor valor de GMeanLoss significa dizer que os valores obtidos com a imputação, fornecidos pela rede geradora, diferem pouco do valor dos dados reais, caracterizando assim uma melhor qualidade da imputação com a adoção dos hiperparâmetros que possuem este respectivo valor de GMeanLoss.

Tabela 4 – Melhores Hiperparâmetros: rede GAIN aprimorada

Hiperparâmetro	**Valor**
Número de camadas ocultas	3
Taxa de dicas (*Hint*)	0,603758
Número de épocas (*Epoch*)	19
Taxa de aprendizado (*Learning Rate*)	0,013733
Número de neurônios nas camadas ocultas	[66, 255,344, 289, 88 e 298]
Tamanho do lote (*Batch Size*)	128
Funções de ativação (*Activation Function*)	[*softsign, tanh, LeakyReLU, tanh, softplus e swish*]
Otimizador (*Optimizer*)	*Adam*

Após a validação, os hiperparâmetros da Tabela 4, foram utilizados no conjunto de teste. Nesse conjunto, conforme dito na seção 4.4, a inserção de lacunas foi feita em percentuais de 5, 10, 20, 30, 40 e 50% e então observado o desempenho frente as métricas MAE e RMSE, Figura 2 e Figura 3, respectivamente. Na seção 5.2 a seguir, será abordada a

fase de teste, buscando avaliar a robustez do GAIN frente ao treinamento com lacunas com a adoção dos hiperparâmetros encontrados através do *Random Search* após a validação.

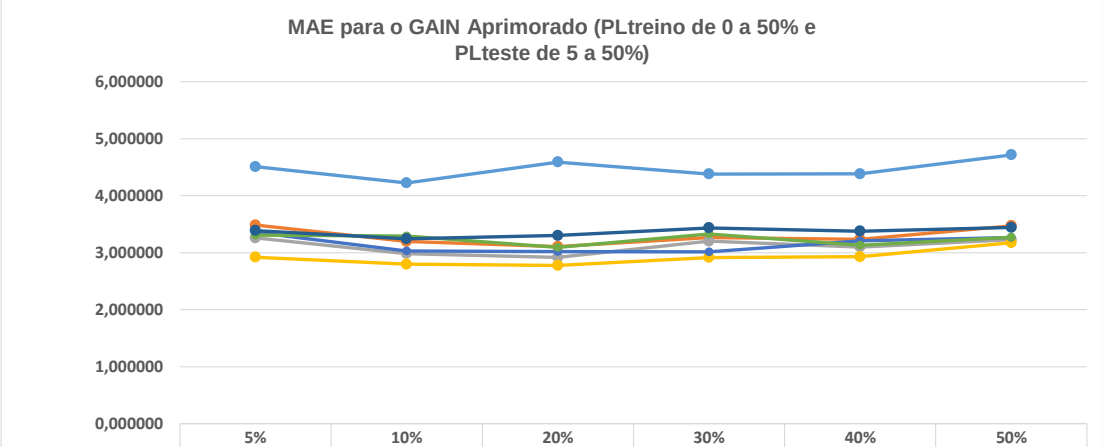
5.2 Robustez do GAIN ao treinamento com lacunas

A robustez de um algoritmo refere-se à sua capacidade de lidar eficazmente com diferentes situações, dados de entrada variados e perturbações. Um algoritmo robusto é capaz de produzir resultados confiáveis e precisos mesmo diante de variações nos dados ou condições adversas. Isso envolve evitar erros ou falhas drásticas, mantendo um desempenho consistente em uma variedade de cenários.

O experimento buscou avaliar o impacto que as lacunas causam no treinamento e teste do método GAIN. Assim, o método GAIN foi submetido a diferentes cenários de treinamento com uma quantidade variável de lacunas, essa avaliação é particularmente importante para se conhecer as limitações que o algoritmo GAIN apresenta para imputar dados diante de cenários com maior número de lacunas.

Cada rede treinada foi submetida à treinos com variação de percentual de lacunas de 0 a 50% (*PLtreino*) e a de testes (*PLteste*) variando o percentual de lacunas de 5 a 50%, buscando avaliar seu desempenho em termos das métricas MAE e RMSE, como mostram as Figuras 2 e 3.

Figura 2 – MAE: GAIN Aprimorado (PLtreino de 0 a 50% e PLteste de 5 a 50%)



	5%	10%	20%	30%	40%	50%
mae_gain_0%	4,508550	4,222730	4,590660	4,378210	4,380910	4,714240
mae_gain_05%	3,484180	3,196610	3,106360	3,266900	3,235310	3,470170
mae_gain_10%	3,259450	2,983550	2,917110	3,200970	3,097840	3,227320
mae_gain_20%	2,919270	2,797110	2,775910	2,915170	2,927120	3,173970
mae_gain_30%	3,355640	3,027190	3,019750	3,014260	3,207100	3,266060
mae_gain_40%	3,305820	3,294210	3,092520	3,328300	3,127340	3,273270
mae_gain_50%	3,386730	3,243220	3,301030	3,435400	3,376220	3,442730

Fonte: Próprio autor.

Na Figura 2, as linhas representam o percentual de lacunas adotado para treinamento e as colunas indicam o percentual de lacunas nos testes. Ao variar-se o *PLteste* de

0 a 50%, observa-se as melhores métricas de MAE para o cenário com 20% de lacunas no conjunto de treino, independente do percentual de lacunas adotado para teste (linha amarela). Ao passo que obtém-se os valores mais altos de MAE quando *PLteste* é 0%, ou seja, quando o conjunto é treinado sem a presença de lacunas.

Valores mais altos no cenário com *PLteste* igual 0%, pode estar relacionado ao fato de que ao se treinar com um conjunto de dados sem lacunas pode se tornar desafiador para a rede conseguir generalizar corretamente, refletindo assim no aumento das métricas de erro como fora observado nesse cenário.

No cenário com melhor valor de MAE (20% de lacunas no conjunto de teste), observa-se que o algoritmo alcança seu menor erro médio absoluto também com o valor de 20% de lacunas para o *PLteste* (2,77ºC). Embora, observe-se uma variação na métrica MAE ao variar-se o *PLtreino*, ainda assim essa variação não ultrapassa mais que aproximadamente 1°C para todas as variações no *PLtreino*, com exceção se compararmos com o conjunto treinado sem a presença de lacunas, onde a diferença chega em aproximadamente 1,6°C, mas ainda assim não representa uma variação significativa na métrica, dando indícios da robustez do algoritmo GAIN frente a variação no percentual de lacunas de treino.

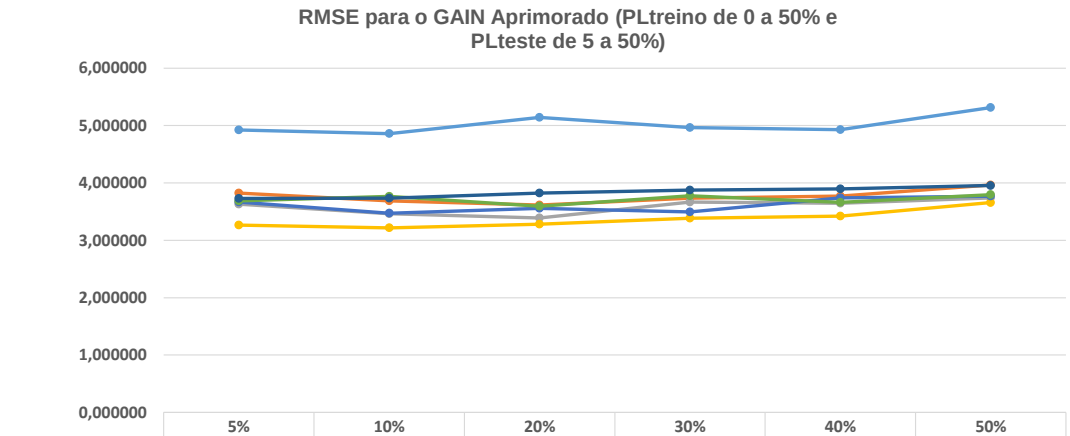
Sob a perspectiva da variação do percentual de lacunas de teste *(PLteste)* de 5 a 50% percebe-se que o valor da métrica MAE para cada um dos conjuntos de treinamento não varia substancialmente. Na linha com menor *PLtreino* (20%), por exemplo, o valor da métrica MAE varia entre os extremos de 2,77 (*PLteste* = 20%) e 3,17 (*PLteste* = 50%), uma variação na métrica em análise de apenas 0,4ºC, a proximadamente. Esse comportamento se repete mesmo para o pior cenário; quando o *PLtreino* é 0%, a variação máxima no valor da métrica MAE é de aproximadamente 0,49°C, sendo obtida pela diferença entre os valores de 4,71°C para *PLteste* = 50% e 4,22ºC para *PLteste* = 10% ilustrados na figura 2.

Vale ressaltar que esse aumento do erro, mesmo que mínimo, é esperado porque a cada variação de percentual de lacunas, o método diminui a probabilidade de acerto, visto que a quantidade de dados que o ajudaria a definir as características do dado a ser imputado está menor. Contudo, ainda assim o algoritmo performa bem, variando minimamente frente as mudanças no cenário de quantitativo de lacunas tanto no treino quanto no teste ao se inserir lacunas no conjunto de dados.

Buscou-se calcular também a métrica RMSE, conforme Figura 3. Nessa figura, varia-se o percentual de lacunas nos conjuntos de dados de treino e de teste de 0 a 50% e de 5 a 50%, respectivamente. Nela, se comparado a Figura 2, observa-se um pequeno aumento no valor da métrica, esse comportamento se dá devido a natureza quadrática do cálculo do RMSE que leva em conta os erros quadráticos entre as previsões e os valores reais. Isso significa que os erros maiores são penalizados mais fortemente. Contudo, a robustez do algoritmo GAIN frente as variações de lacunas existentes no conjunto de

dados é reafirmada, tendo em vista que mesmo aumentando o percentual de lacunas, o GAIN com a arquitetura proposta não apresenta diferenças consideráveis entre os valores esperado e previsto.

Figura 3 – RMSE: GAIN Aprimorado (PLtreino de 5 a 50% e PLteste de 5 a 50%)



	5%	10%	20%	30%	40%	50%
rmse_gain_0%	4,925370	4,859280	5,142230	4,965970	4,930470	5,314340
rmse_gain_05%	3,823940	3,687020	3,615040	3,733900	3,770650	3,964880
rmse_gain_10%	3,632600	3,461690	3,389600	3,666640	3,645860	3,739020
rmse_gain_20%	3,266640	3,216620	3,279230	3,384500	3,422830	3,658750
rmse_gain_30%	3,667490	3,471820	3,559360	3,497440	3,741210	3,768670
rmse_gain_40%	3,689330	3,764930	3,594390	3,778390	3,663310	3,795470
rmse_gain_50%	3,730030	3,732960	3,821530	3,875720	3,897450	3,956360

Fonte: Próprio autor.

Na Figura 3, observa-se que ao se variar o percentual de lacunas de treino de 0 à 50%, o menor valor de erro, nesse caso para o RMSE, permanece quando se adota um *PLteste* de 20%. Nessa figura, fica um pouco mais evidente uma pequena tendência de crescimento linear do erro médio quadrático quando se varia o percentual de lacunas no teste de 5 à 50%. Para o melhor cenário observado na Figura 3 (PLtreino = 20%), onde observa-se nos valores extremos de PL adotados neste estudo um aumento de 2,91°C para 3,17°C, respectivamente. Uma variação de apenas 0,26°C.

Contudo, ainda que observe-se um crescimento linear nos valores da métrica, essa evolução do erro diante de uma variação do percentual de lacunas no teste de até metade do valor do conjunto de dados não é significativa e todas as curvas, independente do percentual de lacunas adotado durante o treino, seguem essa tendência. Isso demonstra a robustez a variação de lacunas.

5.3 Comparação do GAIN com KNN e MICE

Esta seção traz uma avaliação comparativa em termos das métricas MAE e RMSE de algoritmos que utilizam redes adversárias generativas (GAIN e GAIN aprimorado) e algoritmos mais tradicionais de imputação de dados (KNN e MICE). Na Figura 4, o

desempenho dos algoritmos supracitados para um treinamento com 20% de lacunas, pode ser observado.

Na Figura 4, as barras laranja, cinza, amarela e azul representam o desempenho dos algoritmos KNN, MICE, GAIN aprimorado e GAIN, respectivamente. Na figura, observa-se que há uma variação mínima do erro médio absoluto ao se variar o percentual de lacunas no teste de 5 à 50% para o caso dos algoritmos que usam redes adversárias GAIN, isso significa que mesmo que metade do conjunto de dados esteja faltante ainda assim a qualidade das imputações se mantém. No caso dos algoritmos KNN e MICE percebe-se que o valor do MAE aumenta linearmente a medida que o *PLteste* cresce de 5 à 50%.

Para o KNN, por exemplo, quando eleva-se de 5 para 50% o valor do *PLteste*, o MAE varia em 0,83ºC, aproximadamente. Para o algoritmo MICE, o valor do MAE passa de 0,63 ($PLteste = 5\%$) para 1,54 ($PLteste = 50\%$), uma variação de 0,91ºC. Para a rede GAIN (barra azul), essa variação é um pouco menor, 0,34ºC e para a rede GAIN aprimorada (barra amarela), a variação no valor do MAE para um *PLteste* de 5 para 50% é de 0,26ºC.

Um ponto interessante observado é a melhoria observada no valor da métrica MAE quando adota-se a rede GAIN aprimorada, que faz uso do Random Search e adição de mais uma camada oculta. Essa configuração se comparada àquela que utiliza os hiperparâmetros baseados no trabalho de (YOON; JORDON; SCHAAR, 2018) apresentou de fato uma melhoria da qualidade das imputações tendo em vista a redução do valor do MAE.

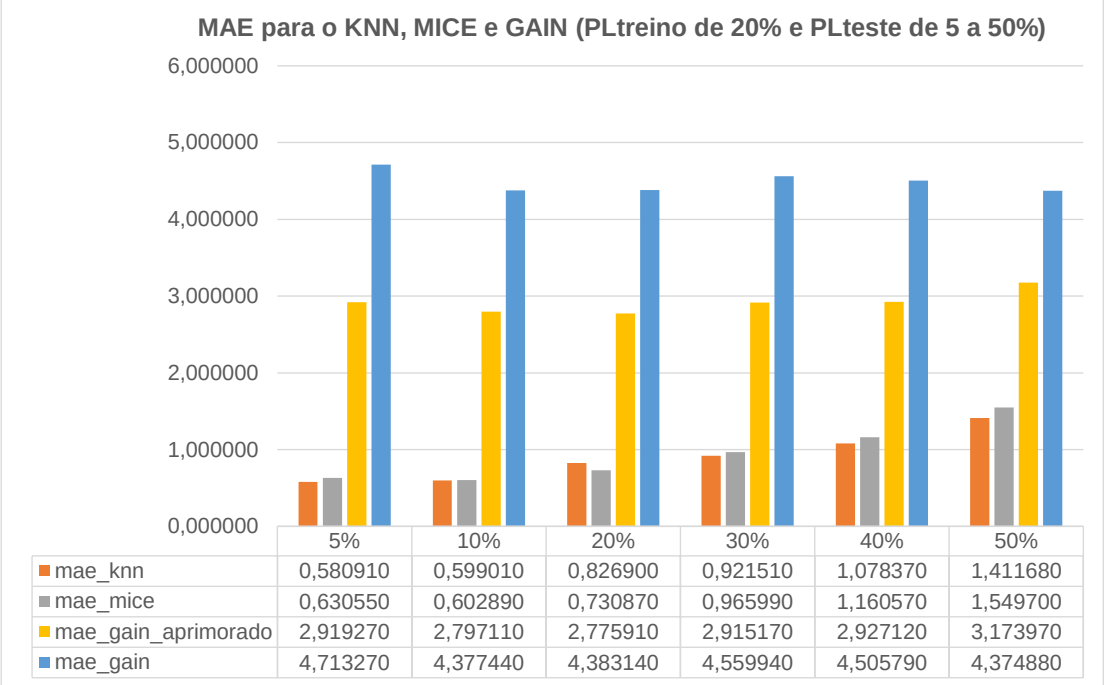
Com *PLteste* de 40%, por exemplo, observa-se que o GAIN aprimorado apresenta um MAE de 2,91ºC ao passo que a rede GAIN proposta por (YOON; JORDON; SCHAAR, 2018), apresenta um MAE de 4,5ºC. Essa melhoria no valor da métrica ressalta a influência do uso de ferramentas de otimização como o *Random Search* para escolha de hiperparâmetros e melhoria na qualidade de imputações em casos de redes adversárias generativas.

Apesar do uso de ferramentas como o *Random Search* para redes GAIN melhorar significativamente seu desempenho em termos da métrica MAE como já fora observado, ainda assim quando comparadas aos algoritmos KNN e MICE, embora sejam métodos mais simples, estes últimos apresentaram um menor valor na métrica MAE, demonstrando que para o conjunto de dados utilizado e suas características, o KNN e MICE seriam mais indicados.

Outra métrica observada para avaliar o desempenho dos algoritmos em apreço nesse estudo foi a RMSE. A Figura 5, apresenta o retorno obtido para essa métrica quando varia-se o *PLteste* de 5 à 50% com *PLtreino* de 20%. Nessa figura, o comportamento se assemelha aos obtidos com o MAE, onde o KNN e MICE apresentam menores valores na métrica, ou seja, o melhor desempenho na qualidade das imputações para este estudo,

variando pouco e de maneira linear a medida que aumenta-se o percentual de lacunas no teste, ao passo que, os algoritmos que utilizaram redes generativas adversárias mantiveram um comportamento constante a medida que varia-se o *PLteste*, onde seu valor de RMSE se manteve entre 2,9 e 3,17ºC para o GAIN aprimorado e entre 4,37 e 4,71ºC para o GAIN de base, ou seja, àquele que não fez uso de hiperparâmetros otimizados pelo *Random Search*.

Figura 4 – MAE para o KNN, MICE e GAIN (PLtreino de 20% e PLteste de 5 a 50%)



	5%	10%	20%	30%	40%	50%
mae_knn	0,580910	0,599010	0,826900	0,921510	1,078370	1,411680
mae_mice	0,630550	0,602890	0,730870	0,965990	1,160570	1,549700
mae_gain_aprimorado	2,919270	2,797110	2,775910	2,915170	2,927120	3,173970
mae_gain	4,713270	4,377440	4,383140	4,559940	4,505790	4,374880

Fonte: Próprio autor.

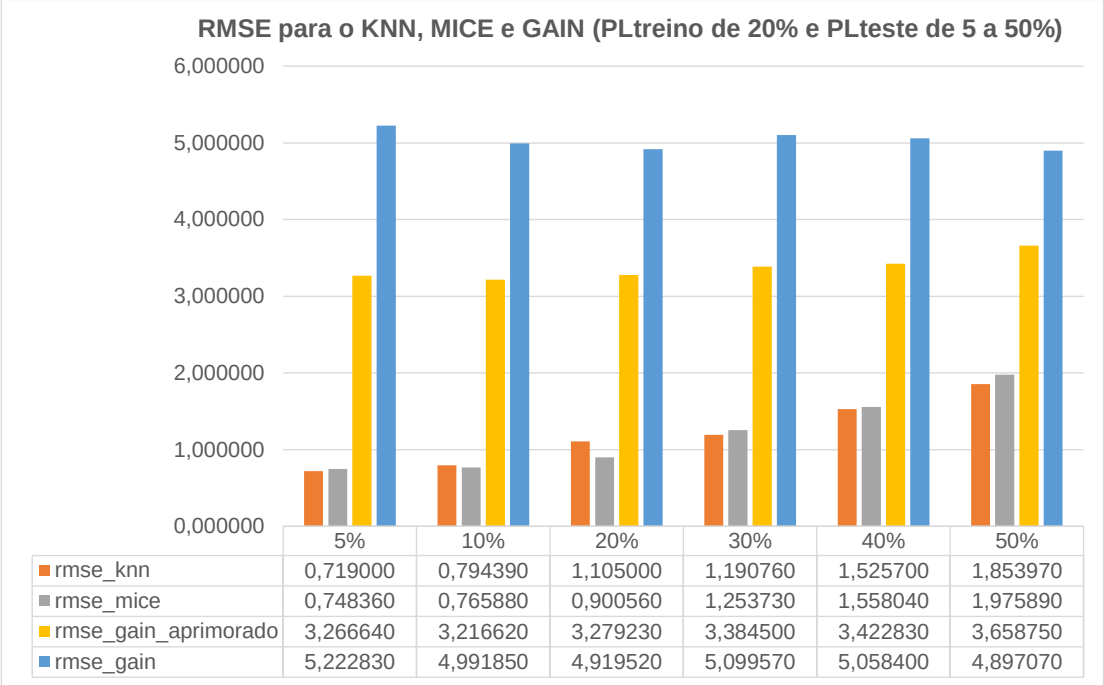
Tanto na Figura 4 quanto na Figura 5, os algoritmos KNN e MICE demonstram ser equiparáveis em termos do desempenho em relação às métricas, retornando valores semelhantes em todos os *PLteste* observados e apresentando melhor desempenho que o GAIN de base e o GAIN aprimorado.

Ao comparar-se o GAIN de base com o GAIN aprimorado observa-se uma diferença notória em seu desempenho. Enquanto o GAIN aprimorado apresenta MAE de 2,9ºC e RMSE 3,26ºC (*(PLteste* = 5%), o GAIN de base apresenta 4,71ºC para o MAE e 5,22ºC para o RMSE. Assim, nota-se que mesmo com o conjunto de dados trabalhando com um percentual mínimo de lacunas (5%), o GAIN de base trás uma performance menor que o GAIN aprimorado, ou seja, a diferença entre o valor real e o valor imputado com o uso desse algoritmo sem o uso de ferramentas como o *Random Search* é alta.

Os achados observados aqui mostram que trabalhar com imputação de dados é algo complexo, e o desempenho de algoritmos utilizados para esse fim dependem de diversos fatores que influenciam diretamente na aderência dos dados ao modelo de imputação

adotado. Por isso a importância dessa análise, onde constatou-se que métodos mais simples podem ser eficazes e úteis para esse tipo de problema e que técnicas de otimização como o *Random Search* podem ser importantes aliadas para melhoria do desempenho de redes adversárias generativas em problemas com níveis de complexidade acentuados.

Figura 5 – RMSE para o KNN, MICE e GAIN (PLtreino de 20% e PLteste de 5 a 50%)



	5%	10%	20%	30%	40%	50%
rmse_knn	0,719000	0,794390	1,105000	1,190760	1,525700	1,853970
rmse_mice	0,748360	0,765880	0,900560	1,253730	1,558040	1,975890
rmse_gain_aprimorado	3,266640	3,216620	3,279230	3,384500	3,422830	3,658750
rmse_gain	5,222830	4,991850	4,919520	5,099570	5,058400	4,897070

Fonte: Próprio autor.

6 Conclusões

Este trabalho investigou o comportamento de redes adversárias generativas para imputação (GAIN) sobre dados de temperatura máxima multivariados e espacialmente distribuídos. Em particular, o trabalho apresentou resultados da influência da quantidade de lacunas pré-existentes nos dados no treinamento de redes GAIN e realizou uma análise comparativa do desempenho dessas redes frente à algoritmos clássicos de aprendizagem de máquina.

Nesse estudo, observou-se que os algoritmos clássicos demonstraram um melhor desempenho nas métricas MAE e RMSE ao variar-se o percentual de lacunas no teste, se comparado aos algoritmos que utilizam redes adversárias generativas. Este resultado deve ser destacado, pois os modelos gerados pelo GAIN são complexos e tendem a ter um grande número de parâmetros (pesos) e hiperparâmetros para ser otimizado, tornando-os fortemente dependente da quantidade de dados de entrada e da necessidade de processos cuidadosos de busca de hiperparâmetros. Tal fato foi evidenciado neste trabalho, pois observou-se a melhoria do desempenho nas métricas MAE e RMSE das imputações da rede GAIN quando adotou-se os melhores hiperparâmetros apontados com o uso do método de otimização *Random Search* e a adição de mais uma camada oculta capaz de aumentar a complexidade do modelo, permitindo-lhe aprender características mais profundas e melhorar o seu desempenho na tarefa de imputação de lacunas.

Um outro ponto constatado foi o fato de que os algoritmos GAIN apresentaram robustez, com variações mínimas no MAE e RMSE, quando o percentual de dados ausentes variou de 5% a 50%. Isso sugere que, apesar de não ter apresentado o melhor desempenho se comparado ao KNN e MICE, o valor das imputações permaneceu consistente, mesmo com um grande volume de lacunas.

Baseado nos resultados deste estudo, o GAIN pode ser usado para problemas que envolvam um maior percentual de lacunas e que apresentem dados com características complexas e de não-linearidade a depender do problema em análise e dos valores que o MAE e RMSE possam assumir para esse respectivo problema. De mesmo modo, o uso de técnicas de imputação de dados como KNN e MICE podem ser suficientes a depender da situação em análise. É necessário portanto a análise do contexto de aplicação do algoritmo e demais nuances da situação problema observada e suas particularidades para a escolha do modelo mais adequado.

6.1 Contribuições

Este trabalho visou avaliar o impacto da taxa de falhas no treinamento de redes adversárias generativas para imputação de dados de temperatura multivariados e espacialmente distribuídos da APAC (Agência Pernambucana de Águas e Clima), onde observou-se que a solução se demonstrou robusta mesmo em meio ao aumento do percentual de lacunas no conjunto de dados, além disso, apresentou uma melhoria substancial nas imputações no contexto apresentado, quando teve seus hiperparâmetros otimizados pela técnica de otimização de hiperparâmetros *Random Search*.

Outras contribuições dessa dissertação são:

- Desenvolvimento de um sistema para coleta de dados da APAC, que pode ser utilizado em experimentos e aplicações semelhantes (os dados coletados estão sendo armazenados na platarforma Thingsboard disponível na UFRPE, como comentado na seção 4.1);
- Aprimoramento do modelo GAIN inicial utilizando a técnica de otimização de hiperparâmetros *Random Search*;

6.2 Trabalhos futuros

Como sugestões de trabalhos futuros tem-se a expansão do estudo para dados a nível Brasil, possibilitando assim a experimentação e análise de uma distribuição de dados com ainda mais variedade e com maior quantidade de dados. Sugere-se ainda investir em infraestrutura de treinamento para aumento da capacidade de processamento dos dados durante as fases de treino e teste bem como aprimorar o uso do *Random Search* para definição de hiperparâmetros de algoritmos que envolvam maior complexidade, como é o caso do GAIN.

Referências

ABADI, M. et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*. 2015. <https://www.tensorflow.org/>. Citado 2 vezes nas páginas 28 e 33.

ADDI, M. et al. Evaluation of imputation techniques for infilling missing daily rainfall records on river basins in ghana. *Hydrological Sciences Journal*, Taylor Francis, v. 67, n. 4, p. 613–627, 2022. Disponível em: <https://doi.org/10.1080/02626667.2022.2030868>. Citado na página 29.

ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. *Wasserstein GAN*. 2017. Citado na página 17.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, v. 13, n. Feb, p. 281–305, 2012. Citado na página 33.

BEZERRA, D. et al. Availability assessment of weather measurement stations. In: SBC. *Anais do X Workshop de Computação Aplicada a Gestão do Meio Ambiente e Recursos Naturais*. [S.l.], 2019. p. 1–10. Citado na página 1.

BLEIDORN, M. T. et al. Methodological approaches for imputing missing data into monthly flows series. *Revista Ambiente & Água*, SciELO Brasil, v. 17, 2022. Citado na página 6.

CAO, W. et al. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, v. 31, 2018. Citado na página 24.

CARVALHO, J. R. P. de et al. Model for multiple imputation to estimate daily rainfall data and filling of faults. *Revista Brasileira de Meteorologia*, SciELO Brasil, v. 32, p. 575–583, 2017. Citado 2 vezes nas páginas 23 e 26.

CHAI, T.; DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, Copernicus Publications Göttingen, Germany, v. 7, n. 3, p. 1247–1250, 2014. Citado na página 22.

CHE, Z. et al. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, Nature Publishing Group UK London, v. 8, n. 1, p. 6085, 2018. Citado na página 24.

CHOLLET, F. *Keras*. 2015. <https://keras.io/>. Citado 2 vezes nas páginas 28 e 33.

DIOUF, S.; DÈME, A. et al. Imputation methods for missing values: the case of senegalese meteorological data. *African Journal of Applied Statistics*, Statistics and Probability African Society, v. 9, n. 1, p. 1245–1278, 2022. Citado 5 vezes nas páginas 1, 2, 10, 11 e 13.

ELY, D. F. et al. Análise de métodos para o preenchimento de falhas aplicados em séries de dados pluviométricos do estado do paraná (brasil). *Raega-O Espaço Geográfico em Análise*, v. 51, p. 122–142, 2021. Citado na página 5.

EMMANUEL, T. et al. A survey on missing data in machine learning. *Journal of Big Data*, SpringerOpen, v. 8, n. 1, p. 1–37, 2021. Citado 2 vezes nas páginas 5 e 7.

ENDERS, C. K. *Applied missing data analysis*. [S.l.]: Guilford Publications, 2022. Citado na página 7.

FERRERO, C. A. et al. Previsao da temperatura da água no reservatório de itaipu utilizando o método nao-linear k-nearest neighbor. In: *III Congresso da Academia Trinacional de Ciências, Foz do Iguaçu—PR, Brasil*. [S.l.: s.n.], 2008. p. 1–10. Citado na página 10.

FLORES, A.; TITO, H.; CENTTY, D. Model for time series imputation based on average of historical vectors, fitting and smoothing. *International Journal of Advanced Computer Science and Applications*, Science and Information (SAI) Organization Limited, v. 10, n. 10, 2019. Citado na página 1.

FLORES, A.; TITO, H.; CENTTY, D. Recurrent neural networks for meteorological time series imputation. *International Journal of Advanced Computer Science and Applications*, Science and Information (SAI) Organization Limited, v. 11, n. 3, 2020. Citado 2 vezes nas páginas 5 e 36.

FLORES, A.; TITO, H.; SILVA, C. Local average of nearest neighbors: Univariate time series imputation. *International Journal of Advanced Computer Science and Applications*, Science and Information (SAI) Organization Limited, v. 10, n. 8, 2019. Citado na página 1.

GOODFELLOW, I. *NIPS 2016 Tutorial: Generative Adversarial Networks*. 2017. Citado na página 17.

GOODFELLOW, I. J. et al. *Generative Adversarial Networks*. 2014. Citado na página 15.

HASANOV, M.; WOLTER, M.; GLENDE, E. Time series data splitting for short-term load forecasting. In: *PESS + PELSS 2022; Power and Energy Student Summit*. [S.l.: s.n.], 2022. p. 1–6. Citado na página 35.

HUDAK, A. T. et al. Nearest neighbor imputation of species-level, plot-scale forest structure attributes from lidar data. *Remote Sensing of Environment*, Elsevier, v. 112, n. 5, p. 2232–2245, 2008. Citado na página 24.

JING, X. et al. A multi-imputation method to deal with hydro-meteorological missing values by integrating chain equations and random forest. *Water Resources Management*, Springer, v. 36, n. 4, p. 1159–1173, 2022. Citado 6 vezes nas páginas 2, 4, 5, 10, 29 e 36.

KIANI, K.; SALEEM, K. K-nearest temperature trends: A method for weather temperature data imputation. In: *Proceedings of the 2017 International Conference on Information System and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2017. (ICISDM '17), p. 23–27. ISBN 9781450348331. Disponível em: <https://doi.org/10.1145/3077584.3077592>. Citado na página 11.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: *3rd International Conference on Learning Representations (ICLR)*. [s.n.], 2014. Disponível em: <https://arxiv.org/abs/1412.6980>. Citado na página 32.

LITTLE, R. J.; RUBIN, D. B. *Statistical Analysis with Missing Data*. [S.l.]: John Wiley Sons, 2019. Citado 6 vezes nas páginas 5, 6, 7, 8, 9 e 21.

LUO, Y. et al. E2gan: End-to-end generative adversarial network for multivariate time series imputation. In: AAAI PRESS. *Proceedings of the 28th international joint conference on artificial intelligence*. [S.l.], 2019. p. 3094–3100. Citado 5 vezes nas páginas 2, 11, 24, 26 e 29.

MENG, X. et al. A novel deep learning-based robust dual-rate dynamic data modeling for quality prediction. *IEEE Transactions on Industrial Informatics*, p. 1–11, 2023. Citado na página 33.

MIR, A. A. et al. An improved imputation method for accurate prediction of imputed dataset based radon time series. *Ieee Access*, IEEE, v. 10, p. 20590–20601, 2022. Citado 5 vezes nas páginas 1, 4, 5, 10 e 36.

MORAES, R. A.; ARRAES, C. L. Análise de uma metodologia para preenchimento de valores faltantes em dados de precipitação, para o estado do paraná. *UNOPAR Científica Ciências Exatas e Tecnológicas*, v. 11, n. 1, 2012. Citado na página 11.

MORUP, M. et al. *Scalable tensor factorizations with missing data*. [S.l.], 2010. Citado na página 24.

NUNES, L. N.; KLÜCK, M. M.; FACHEL, J. M. G. Comparação de métodos de imputação única e múltipla usando como exemplo um modelo de risco para mortalidade cirúrgica. *Revista Brasileira de Epidemiologia*, SciELO Public Health, v. 13, n. 4, p. 596–606, 2010. Citado na página 5.

OLIVEIRA, L. Á. et al. Mineração de regras de associação diversas em dados meteorológicos temporais de múltiplos pontos geográficos via algoritmo genético. *Brazilian Journal of Development*, v. 7, n. 12, p. 112027–112048, 2021. Citado 2 vezes nas páginas 23 e 26.

OLIVEIRA, L. M. de et al. Multiple imputation to fill in missing data in soil physico-hydrical properties database. *Revista Ciência Agronômica*, v. 51, n. 4, p. 1–10, 2020. Citado 3 vezes nas páginas 12, 13 e 14.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 28.

PEURIFOY, J. E. *The physics of artificial intelligence*. Tese (Doutorado) — Massachusetts Institute of Technology, 2018. Citado na página 24.

POPOLIZIO, M. et al. The gain method for the completion of multidimensional numerical series of meteorological data. *IAENG International Journal of Computer Science*, v. 48, n. 3, 2021. Citado 5 vezes nas páginas 2, 20, 24, 25 e 26.

RUBIN, D. B. Inference and missing data. *Biometrika*, Oxford University Press, v. 63, n. 3, p. 581–592, 1976. Citado 5 vezes nas páginas 7, 8, 9, 13 e 14.

RUBIN, D. B. *Multiple Imputation for Nonresponse in Surveys*. [S.l.]: Wiley, 1987. 258 p. Citado na página 6.

RUBIN, D. B.; SCHENKER, N. Multiple imputation for interval estimation from simple random samples with ignorable nonresponse. *Journal of the American Statistical Association*, [American Statistical Association, Taylor Francis, Ltd.], v. 81, n. 394, p. 366–374, 1986. ISSN 01621459. Disponível em: <http://www.jstor.org/stable/2289225>. Citado na página 12.

SAMAL, K. K. R. et al. An improved pollution forecasting model with meteorological impact using multiple imputation and fine-tuning approach. *Sustainable Cities and Society*, v. 70, p. 102923, 2021. ISSN 2210-6707. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2210670721002092>. Citado 2 vezes nas páginas 2 e 11.

SHAHBAZIAN, R.; TRUBITSYNA, I. Degain: Generative-adversarial-network-based missing data imputation. *Information*, v. 13, n. 12, 2022. ISSN 2078-2489. Disponível em: <https://www.mdpi.com/2078-2489/13/12/575>. Citado 2 vezes nas páginas 18 e 21.

WANG, Y. et al. Pc-gain: Pseudo-label conditional generative adversarial imputation networks for incomplete data. *Neural Networks*, v. 141, p. 395–403, 2021. ISSN 0893-6080. Disponível em: <https://www.sciencedirect.com/science/article/pii/S089360802100229X>. Citado na página 16.

WILLMOTT, C. J.; MATSUURA, K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, v. 30, n. 1, p. 79–82, 2005. Citado na página 22.

WU, R. et al. Data imputation for multivariate time series sensor data with large gaps of missing data. *IEEE sensors journal*, IEEE, v. 22, n. 11, p. 10671–10683, 2022. ISSN 1530-437X. Citado 2 vezes nas páginas 25 e 26.

YANG, S. et al. Adversarial recurrent time series imputation. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, 2020. Citado 5 vezes nas páginas 4, 5, 10, 29 e 36.

YOON, J.; JORDON, J.; SCHAAR, M. Gain: Missing data imputation using generative adversarial nets. In: PMLR. *International conference on machine learning*. [S.l.], 2018. p. 5689–5698. Citado 9 vezes nas páginas 10, 1, 14, 16, 18, 19, 32, 33 e 41.

YOON, S.; SULL, S. Gamin: Generative adversarial multiple imputation network for highly missing data. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2020. p. 8456–8464. Citado 3 vezes nas páginas 17, 18 e 22.

ZHANG, Y. et al. Missing value imputation in multivariate time series with end-to-end generative adversarial networks. *Information Sciences*, Elsevier, v. 551, p. 67–82, 2021. Citado na página 7.