



UNIVERSIDADE FEDERAL DO MARANHÃO
Programa de Pós-Graduação em Ciência da Computação

Mauricio Morais Almeida

***Imputação de Dados Faltosos em Séries Temporais Univariadas
utilizando Meta-Aprendizado baseado em Rede Neural LSTM
Híbrida***

São Luís
2023

MAURICIO MORAIS ALMEIDA

**IMPUTAÇÃO DE DADOS FALTOSOS EM SÉRIES TEMPORAIS UNIVARIADAS
UTILIZANDO META-APRENDIZADO BASEADO EM REDE NEURAL LSTM
HÍBRIDA**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFMA como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. João Dallyson Sousa de Almeida
Coorientador: Prof. Dr. Darlan Bruno Pontes Quintanilha

SÃO LUÍS – MA

2023

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Morais Almeida, Mauricio.

Imputação de Dados Faltosos em Séries Temporais
Univariadas utilizando Meta-Aprendizado baseado em Rede
Neural LSTM Híbrida São Luís 2023 / Mauricio Morais
Almeida. - 2023.

96 p.

Coorientador(a): Darlan Bruno Pontes Quintanilha.

Orientador(a): João Dallyson Sousa de Almeida.

Dissertação (Mestrado) - Programa de Pós-graduação em
Ciência da Computação/ccet, Universidade Federal do
Maranhão, São Luis-MA, 2023.

1. HybridLSTM. 2. Imputação de dados. 3. Meta-
aprendizado. 4. Pix2Pix. 5. Séries Temporais. I. Pontes
Quintanilha, Darlan Bruno. II. Sousa de Almeida, João
Dallyson. III. Título.

MAURICIO MORAIS ALMEIDA

**IMPUTAÇÃO DE DADOS FALTOSOS EM SÉRIES TEMPORAIS UNIVARIADAS
UTILIZANDO META-APRENDIZADO BASEADO EM REDE NEURAL LSTM
HÍBRIDA**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFMA como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Dissertação SÃO LUÍS – MA, 5 de maio de 2023.

Prof. Dr. João Dallyson Sousa de Almeida
Orientador

Prof. Dr. Darlan Bruno Pontes Quintanilha
Coorientador

Prof. Dr. João Otávio Bandeira Diniz
(Avaliador Interno)

Prof. Dr. Ginalber Luiz de Oliveira Serra
(Avaliador Externo)

SÃO LUÍS – MA
2023

A Minha família de heróis, minha esposa, meus pais, meus irmãos e meus sogros.

AGRADECIMENTOS

A Deus, minha força, inspiração e companheiro nos bons e maus momentos. À minha esposa, que nunca me desamparou e nunca descreditou de meus propósitos, tonou-se mais que necessária na minha jornada. A meus pais e irmãos por sempre apoiar e torcerem por mim, meu refúgio e fonte de alegria.

Ao meu orientador, João Dallyson, uma fonte de conhecimento, exemplo de humildade e admiração, suas orientações de cunho profissional e pessoal vão além do dever, isso o torna mais que merecedor de todas suas conquistas, tenho certeza que sua paciência, sabedoria, conhecimento e reconhecimento do fator humano nas relações profissionais torna sua presença mais que desejável em qualquer ambiente.

Também ao meu coorientador Darlan Quintanilha, que chegou um pouco mais tarde para me ajudar nessa caminhada, mas que já pude perceber que bebe na mesmo fonte que Dallyson, pois possui notório saber. Sua paciência e simplicidade também torna sua presença desejável e seu discurso palatável.

Aos meus novos amigos que sempre me ajudaram e compreenderam que juntos estamos na busca de mais conhecimento, com eles pude confirmar que ninguém faz nada sozinho.

A todos os professores do mestrado, obrigada pelo conhecimento passado e pela inspiração. Obrigada por ensinarem com paixão e dedicação.

Agradeço também a CAPES, pois o presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 "This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001"

Por fim, obrigado a todos que me ajudaram e não percebi, sei que em todos os lugares sempre têm alguém que estende suas mãos sem esperar nada em troca.

O temor do Senhor é o princípio do saber, mas os loucos desprezam a sabedoria e o ensino. Provérbios 1:7

RESUMO

Séries temporais são dados coletado ao longo do tempo regularmente, descrevendo a média de um evento no tempo. Por esse, e outros motivos, as séries temporais vêm ganhando cada vez mais espaço em diversas áreas, tais como aplicações comerciais, naturais, médicas. Uma das principais problemáticas envolvendo séries temporais está na perda de dados e, para recuperá-los, existem diversas abordagens de imputação em séries temporais univariadas. Com objetivo de contribuir com a área de imputação em séries temporais, este estudo propõe um novo método de imputação de valores faltosos baseado em meta-aprendizado. Inicialmente, selecionou-se dez técnicas clássicas para imputar dados de séries temporais e a partir do erro construiu-se uma base de metadados, com as séries rotuladas em dez classes, conforme o menor erro obtido. Além das dez técnicas utilizadas, propôs-se uma nova técnica de imputação usando a rede Pix2Pix GAN, que imputa a partir de imagens de séries temporais. Somado a isso, foi proposta uma nova arquitetura de rede denominada HybridLSTM para recomendar, a partir dos metadados rotulados, a melhor técnica de imputação para uma determinada série. Assim, mostrou-se que a rede HybridLSTM sugeriu as melhores técnicas de imputação de dados a partir das características das séries, superando em diversas oportunidades as imputações de técnicas clássicas como interpolação linear e interpolação Akima. A técnica de imputação proposta foi avaliada em nove *datasets* diferentes e alcançou um ASMAPE médio de 9,51%, um máximo de 22,75% e um mínimo de 3,73%. Mostrou-se ainda que a abordagem de imputar dados por meio de janelamento utilizando várias técnicas em pequenas fatias de séries temporais é um campo promissor e, assim, abriu-se espaço para diversas outras pesquisas como a imputação de dados faltosos em séries temporais por meio de imagens e redes GANs.

Palavras-chave: Séries Temporais; Imputação de dados; Meta-aprendizado; Pix2Pix; HybridLSTM.

ABSTRACT

Time series are data collected over time in a regular manner, describing the average of an event over time. For this reason, among others, time series have been gaining increasing importance in various areas, such as business, natural, and medical applications. One of the main challenges involving time series is data loss, and to recover them, there are various approaches to imputing missing values in univariate time series. In order to contribute to the field of imputation in time series, this study proposes a new method of imputing missing values based on meta-learning. Initially, ten classical techniques were selected to impute time series data, and based on the error, a metadata set was constructed with the series labeled into ten classes according to the lowest obtained error. In addition to the ten techniques used, a new imputation technique using the Pix2Pix GAN network was proposed, which imputes based on images of time series. Furthermore, a new network architecture called HybridLSTM was proposed to recommend the best imputation technique for a given series based on the labeled metadata. It was shown that the HybridLSTM network suggested the best data imputation techniques based on the characteristics of the series, surpassing classical techniques such as linear interpolation and Akima interpolation in several instances. The proposed imputation technique was evaluated on nine different datasets and achieved an average ASMAPE of 9.51%, with a maximum of 22.75% and a minimum of 3.73%. It was also shown that the approach of imputing data through windowing using various techniques on small slices of time series is a promising field, opening up space for various other research areas such as imputing missing data in time series through images and GAN networks.

Keywords: Time series, Convolutional Neural Networks, Time Series Image, Meta-Learning, Imputation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Componentes de uma série temporal extraídos da série <i>Air Passengers</i> .	27
Figura 2 – Estrutura de funcionamento de um neurônio artificial.	28
Figura 3 – Arquitetura de uma rede MLP com uma camada oculta.	30
Figura 4 – Estrutura De uma Rede Adversária Generativa	36
Figura 5 – Tradução de usando Pix2Pix	36
Figura 6 – Componentes da rede Pix2Pix	37
Figura 7 – Forma resumida de uma RNN.	40
Figura 8 – Forma estendida de uma RNN.	40
Figura 9 – Rede LSTM padrão	41
Figura 10 – Gráfico das séries com tendência	47
Figura 11 – Gráfico das séries estacionarias	50
Figura 12 – Etapas do método proposto.	52
Figura 13 – Exemplo detalhado do método até a imputação com a série <i>Air Passengers</i>	54
Figura 14 – Etapas do método proposto usando a rede Pix2Pix.	55
Figura 15 – Transformação proposta.	56
Figura 16 – Exemplo de imagem com (<i>Input</i>) e sem (<i>Real</i>) valores faltosos.	56
Figura 17 – Gerador proposto.	57
Figura 18 – Decodificador proposto.	57
Figura 19 – Exemplo detalhado da construção do metadado até a geração dos rótulos	58
Figura 20 – Rede Base para a HybridLSTM	60
Figura 21 – Rede Proposta	61
Figura 22 – Frequência percentual da melhor técnica de imputação definida com base na métrica ASMAPE em 374 séries.	64
Figura 23 – Frequência percentual da melhor técnica de imputação definida com base na métrica ASMAPE em 776 séries.	66
Figura 24 – AUC do treino da rede por execução Holdout no experimento com 10% de valor faltoso.	70
Figura 25 – AUC da rede por execução Holdout no experimento com 20% de valor faltoso.	73
Figura 26 – AUC da rede por execução Holdout no experimento com 30% de valor falso	75

Figura 27 – AUC da rede por execução Holdout no experimento com 10% de valor faltoso em 776 séries.	78
Figura 28 – AUC da rede por execução Holdout no experimento com 20% de valor faltoso em 776 séries.	80
Figura 29 – AUC da rede por execução Holdout no experimento com 30% de valor faltoso em 776 séries.	82

LISTA DE TABELAS

Tabela 1 – Resumo dos trabalhos apresentados	24
Tabela 2 – Principais características dos <i>datasets</i> das séries com tendência	47
Tabela 3 – Principais características dos <i>datasets</i> das séries estacionárias	51
Tabela 4 – Frequência absoluta por técnica de imputação em 5 <i>datasets</i> com base no ASMAPE	64
Tabela 5 – Descrição dos resultados referente a métrica MAE.	65
Tabela 6 – Descrição do desempenho por técnica e dataset referente a métrica ASMAPE	67
Tabela 7 – Descrição da distribuição descritiva da métrica MAE	68
Tabela 8 – Espaço de busca dos melhores parâmetros da rede HyBridLSTM	69
Tabela 9 – Hiperparâmetros do modelo da rede <i>HybridLSTM</i> com melhor desempenho.	69
Tabela 10 – Média do ASMAPE de 5 execuções do Holdout nas primeiras 14 séries com 10% de dados faltosos	71
Tabela 11 – Média do ASMAPE de 374 séries para cada execução Holdout com 10% de dados faltosos por execução Holdout	71
Tabela 12 – Média geral do erro MAE de 374 séries nas imputações com 10% de valor faltoso por execução Holdout	72
Tabela 13 – Média do ASMAPE das 5 execuções Holdout nas primeiras 14 séries com 20% de dados faltosos	74
Tabela 14 – Média geral do erro ASMAPE nas imputações por execução Holdout com 20% de valores faltosos por execução Holdout.	74
Tabela 15 – Média geral do erro MAE nas imputações com 20% de valor faltoso por execução Holdout	75
Tabela 16 – Média do ASMAPE em 374 séries de 5 execuções Holdout nas primeiras 14 séries com 30% de dados faltosos	76
Tabela 17 – Média do ASMAPE de 374 séries de 5 execuções Holdout com 30% de dados faltosos por execução Holdout	76
Tabela 18 – Média geral do erro MAE de 374 séries nas imputações com 30% de valor faltoso por execução Holdout.	77
Tabela 19 – Resultado ASMAPE por <i>Dataset</i> com 10% de valor faltoso.	79

Tabela 20 – Média geral do erro ASMAPE nas imputações por execução Holdout com 10% de valores faltosos.	79
Tabela 21 – Resultado MAE por <i>Dataset</i> com 10% de valor faltoso.	80
Tabela 22 – Resultado ASMAPE por <i>Datatset</i> com 20% de valor faltoso.	81
Tabela 23 – Média do ASMAPE nas Imputações por execução Holdout com 20% de valor faltoso.	81
Tabela 24 – Resultado MAE Por <i>Datatset</i> com 20% de valor faltoso	82
Tabela 25 – Resultado ASMAPE por <i>Dataset</i> com 30% de valor faltoso.	83
Tabela 26 – Média do ASMAPE nas imputações por execução Holdout com 30% de valor faltoso.	83
Tabela 27 – Resultado MAE por <i>Dataset</i> com 30% de valor faltoso.	83
Tabela 28 – Artigo publicado que possui relação com o método proposto.	88

LISTA DE ABREVIATURAS E SIGLAS

AP	<i>Air Passengers</i>
AMM	<i>Ammonia</i>
AR	<i>Aeration rate</i>
ASMAPE	<i>Adapted Symmetric Mean Absolute Percentage Error</i>
BSH	<i>Bike sharing</i>
BS	<i>Beer sales</i>
BP	<i>Parking Birmingham</i>
BN	<i>Batch Normalization</i>
CNN	<i>Convolutional Neural Networks</i>
COVID-19	<i>Corona Virus Disease</i>
DMTME	<i>Daily Minimum Temperatures in ME</i>
EU	<i>Electricity Usage</i>
EP	<i>Electric Production</i>
ELD	<i>Electricity Load Diagrams 20112014</i>
FC	<i>Camadas fully connected</i>
GAN	<i>Generative Adversarial Network</i>
LSTM	<i>Long Short Term Memory</i>
MAE	Erro Absoluto Médio
MARS	<i>Multivariate Adaptive Regression Splines</i>
MBPA	<i>Monthly beer production in Australian</i>
MLP	<i>MultiLayer Perceptron</i>
OMS	Organização Mundial de Saúde

PBW	<i>Paper basis weight</i>
PDI	Processamento Digital de Imagens
RNN	<i>Recurrent Neural Networks</i>
RC	<i>Ruber Colour</i>
SU	<i>Sunspots</i>
STU	Séries Temporais univariadas
STM	Séries Temporais multivariada
SI	Sistema Internacional de Unidades
SVM	<i>Support Vector Machine</i>

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Objetivo Geral	19
1.2	Objetivos Específicos	19
1.3	Organização do Trabalho	20
2	TRABALHOS RELACIONADOS	21
2.1	Considerações Finais	24
3	FUNDAMENTOS TEÓRICOS	26
3.1	Séries Temporais	26
3.2	Redes Neurais	28
3.2.1	Redes de Múltiplas Camadas e <i>Back Propagation</i>	29
3.2.2	Aprendizagem Profunda	33
3.2.3	Redes Neurais Convolucionais	33
3.2.3.1	Camada Convolucional	33
3.2.3.2	Camadas <i>Fully Connected</i> (FC)	34
3.2.3.3	<i>Batch Normalization</i> (BN)	35
3.2.4	Técnica de Imputação Baseada em Aprendizado de Máquina	35
3.3	Técnicas de Clássicas de Imputação	38
3.3.1	Redes <i>Long Short Term Memory</i>	40
3.4	Meta-aprendizado	42
3.5	Métricas de Avaliação	43
4	MATERIAIS E MÉTODO	45
4.1	Aquisição da Base de Dados	45
4.1.1	<i>Datasets</i> com Tendência e Sazonalidade	45
4.1.2	<i>Datasets</i> com Séries Estacionárias	47
4.2	Método Proposto	51
4.2.1	Pré-processamento	52
4.2.2	Imputação	53
4.2.2.1	Técnica de Imputação Utilizando Pix2Pix	55
4.2.3	Construção do Metadado	58

4.2.4	HybridLSTM: Rede proposta para classificação	59
5	RESULTADOS E DISCUSSÃO	62
5.1	Hardware e Softwares Utilizados	62
5.2	Desempenho das Técnicas de Imputação Individuais	62
5.3	Resultados com Meta-Aprendizado	68
5.3.1	Experimento com a HybridLSTM incluindo a rede Pix2Pix	70
5.3.2	Experimento com a HybridLSTM sem a rede Pix2Pix	78
6	CONCLUSÃO	85
6.0.1	Vantagens e Limitações	86
6.1	Contribuições	87
6.2	Trabalhos Futuros	87
6.3	Produções Científicas	88
	REFERÊNCIAS	89

1 INTRODUÇÃO

Séries temporais são conjuntos de observações sequenciais ordenadas pelo tempo (JAVED; LEE; RIZZO, 2020) e, de acordo com Bezerra (2006) podem ser discretas ou contínuas. A principal particularidade dessa sequência é que ela é descrita por um processo estocástico, necessitando de modelagem estatística como andar do bêbado (MLODINOW, 2009), processos autorregressivos (GOMES, 2005), dentre outros. Esses dados estocásticos, têm por essência sua ordenação temporal que permite descrever diversos eventos, como dados econômicos (PATTON, 2012), clínicos (HARUTYUNYAN et al., 2019) e de consumo de energia (CHOU; TRAN, 2018; SEPULVEDA et al., 2021). Aplicações desse tipo de dado geralmente partem de análises baseadas em técnicas estatísticas (LIAO, 2005), como agrupamento, detecção de anomalia (BRAEI; WAGNER, 2020), ou previsão de valores futuros usando métodos como *AutoRegressive Integrated Moving Average* (ARIMA) (FATTAH et al., 2018) e *Prophet* (JHA; PANDE, 2021).

Em termos financeiros, compreender o passado e prever o futuro é crucial para tomar decisões de negócios (ARSHAM, 1994). Em termos sociais, o estudo destes dados permitem entender, interpretar e prever padrões, dentre outros aspectos, em pandemias como a de Covid-19 (ZEROUAL et al., 2020) e em catástrofes naturais (ASLAN et al., 2020).

As séries temporais são classificadas como Séries Temporais Univariadas (STU) ou Séries Temporais Multivariadas (STM) (BRAEI; WAGNER, 2020). As STU possuem apenas uma variável em uma data do tempo, enquanto as multivariadas possuem duas ou mais variáveis em uma mesma data do tempo. Um exemplo de STU é o consumo de água residencial, enquanto os dados necessários para previsão do tempo seriam STM, ao precisar, dentre outras variáveis, da umidade do ar, da hora, e da velocidade do vento.

Quanto a análise dos dados das séries temporais, STU são comumente utilizadas em técnicas computacionais devido sua simplicidade quando comparadas com as STM. No uso de STU, alguns modelos supõem que todos os fatores externos à variável temporal terão um efeito constante sobre o futuro, outros modelos supõem um efeito monótono. Essas suposições geralmente não condizem com a realidade, logo há necessidade de realizar primeiramente a análise de STU, visto que elas representam a versão simplificada do mundo real. Ainda assim, a correlação entre variáveis, o tipo de série e a variação dos

modelos abordados nas séries torna a análise desse tipo de série temporal desafiadora (HAILIN; MIAO, 2020). Por isso, há variações nos modelos de descrição e previsão das séries temporais.

Assim, por conta da aplicabilidade e representatividade inerente à STU, o foco em desenvolvimento e aplicação de modelos matemáticos e de técnicas computacionais para análise dessas séries vão crescendo (KARIM et al., 2019). Com o avanço nessa área, espera-se que técnicas mais eficientes em extrair informação, captar subpadrões e prever valores futuros sejam desenvolvidas e aplicadas, tanto no campo econômico quanto social.

No entanto, para que essas aplicações consigam gerar melhores resultados é necessário possuir dados completos, ou seja, séries temporais sem valores faltando para determinadas datas ou períodos. Contudo, quando se trata de STU reais, a maioria possui dados faltosos (CAILLAULT et al., 2020). Esses dados são perdidos por motivos como armazenamento, falhas em sensores e falhas em transmissão. Para contornar esse problema são realizados diversos estudos para apontar os motivos das perdas de dados (DICKENS; LARSON, 2004), bem como modelos matemáticos para recuperação e imputação de dados faltosos (KHAYATI et al., 2020).

Quando uma série temporal apresenta valores faltantes, esses espaços vazios são preenchidos por meio de modelos de previsão ou de imputação. Entretanto, assim como na previsão, há uma grande variedade de modelos disponíveis para imputação de dados. Essa multiplicidade de técnicas dificulta a escolha do melhor método de imputação, uma vez que não há uma abordagem única que seja superior às demais em todas as séries ou tipos de séries (THOMAS; RAJABI, 2021). Portanto, é necessário o desenvolvimento de abordagens que facilitem a tarefa de escolher o método mais adequado.

A partir do exposto, é inevitável efetuar comparações entre modelos, como observado em (MARTÍNEZ-ÁLVAREZ et al., 2015; MAHALAKSHMI; SRIDEVI; RAJARAM, 2016). Vários estudos, incluindo Junninen et al. (2004), Brock et al. (2008), Yozgatligil et al. (2013), Che et al. (2018), Fang e Wang (2020), indicam que as redes neurais artificiais (RNA) superam modelos consagrados como ARIMA (MOAYEDI; MASNADI-SHIRAZI, 2008) e Média Móvel.

No sentido de facilitar a escolha, Jeon et al. (2021) apontam um caminho mostrando que a partir de meta-aprendizagem que combina três modelos de aprendizado de máquina – *Generalized Linear Model* (GAM), *Gradient Boosting Regressor* (GBR), e

Multivariate Adaptive Regression Splines (MARS) conseguem reduzir os problemas de erros com dados de desempenho de um motor de navio.

Outro ponto positivo do meta-aprendizado está na possibilidade de se construir um método automático de recomendação a partir das características dos dados. Chen et al. (2021) mostra que é possível tanto recomendar como otimizar o processo de recomendação a partir de dados de avaliações e cliques de usuários.

Assim, é possível utilizar o meta-aprendizado como um tipo de análise em séries temporais, principalmente quando não se tem nenhum conhecimento prévio sobre essas séries, para a mineração de informação. Clusterização é uma ferramenta muito útil para mineração em séries temporais, ao permitir encontrar padrões que facilitam a interpretação de um grande conjunto de séries (HALLAC et al., 2017), em essência o meta-aprendizado também realiza o processo de clusterização, pois visa agrupar características conforme o tipo e padrão das séries, ou seja, encontra sua utilização como pré-processamento ou sub-rotina de outras técnicas (PAPARRIZOS; GRAVANO, 2017).

Portanto, levanta-se a hipótese que é possível utilizar o meta-aprendizado para, a partir dos metadados da imputação de dados faltosos, realizados por meio de métodos de imputação em STU, construir um método de recomendação para apontar a melhor técnica de imputação de valores faltosos com base nas características das STU.

1.1 Objetivo Geral

Diante do contexto apresentado, este trabalho tem por objetivo propôr um novo método automático de imputação de dados em STU, usando meta-aprendizado, redes neurais e técnicas clássicas de imputação, que otimize a imputação por meio da recomendação da melhor técnica de imputação de dados faltosos para uma STU.

1.2 Objetivos Específicos

- Construir uma base de dados de STU de diferentes contextos a partir de bases de dados públicas;
- Desenvolver um novo método de imputação de dados por meio de imagens, baseado na rede Pix2Pix GAN;

- Desenvolver um método de recomendação de técnicas de imputação de dados faltosos em STU;
- Incorporar ao método de recomendação técnicas de imputação clássicas, com natureza estatística, e técnicas baseadas em redes neurais;
- Analisar as vantagens e limitações do método proposto com relação à literatura;

1.3 Organização do Trabalho

Este trabalho está organizado da seguinte forma: o **Capítulo 2** apresenta os trabalhos relacionados à temática imputação de dados que serviram como referência para essa dissertação. O **Capítulo 3** apresenta toda a fundamentação teórica, descrevendo os conceitos e técnicas necessárias para o entendimento do método proposto. Este capítulo cobre assuntos como fundamentos de séries temporais, redes neurais, técnicas de imputação e redes neurais utilizadas para imputação, meta-aprendizado, e métricas de avaliação de desempenho na imputação de dados faltosos. O **Capítulo 4** descreve o método proposto nesta pesquisa, além de descrever os materiais necessários para o seu desenvolvimento. Serão abordados tópicos como a construção da base de dados, a criação do metadado e a apresentação da rede proposta. O **Capítulo 5** apresenta, inicialmente, a estatística descritiva dos dados utilizados e dos experimentos realizados somado aos resultados obtidos com o método proposto, bem como os discute e os interpreta. Por fim, o **Capítulo 6** apresenta um resumo do trabalho, evidenciando sua contribuição, tanto para resultados esperados como para resultados impremeditados. Por fim, apresentam-se caminhos para continuidade da pesquisa.

2 TRABALHOS RELACIONADOS

As pesquisas utilizadas como base para esse estudo foram buscadas sobre as palavras-chave: “imputação”, “séries temporais” e “meta-aprendizado”. A faixa de tempo utilizada foi entre 2018 e 2022. Além disso, foram priorizados os estudos desenvolvidos com séries temporais univariadas. Os trabalhos apresentados estão classificados em quatro categorias: (1) comparações para agrupar métodos de imputação ou encontrar padrões no desempenho das imputações, (2) utilização de métodos combinados para imputação, (3) comparação de métodos de imputação em séries temporais com base em aprendizado de máquina em relação aos métodos tradicionais e (4) processos semelhantes à imputação de dados por meio de meta-aprendizado, seja total ou parcialmente.

Bokde et al. (2018) e Afrifa-Yamoah et al. (2020), mostram que a depender de características das STUs, há métodos que por conta dessas características melhor imputam valores ausentes em STUs. Além disso, há um esforço da comunidade científica para propôr com regularidade novos métodos de imputação que em sua maioria superam métodos propostos anteriormente e, isso traz a dificuldade de escolha entre qual método aplicar. Nesse sentido, estão surgindo propostas para se sintetizar esses conhecimentos ou mesmo realizar imputação por meio de meta-aprendizado.

Khayati et al. (2020) avaliaram doze algoritmos de imputação de dados em oito *datasets* de áreas diferentes. O estudo analisou empiricamente esses doze métodos para construir um sistema de recomendação para indicar o melhor método de imputação. A pesquisa levantou pontos fortes, a exemplo, mostra que o desempenho de cada método de imputação pode variar dependendo da série e do tamanho do intervalo de valores faltosos. Somado a isso, pode-se comentar que os autores relacionam as particularidades dos dados, como tendência e sazonalidade, como fatores contribuintes para supremacia de alguns métodos sobre outros. Contudo, os autores realizaram esta análise e, em larga escala, o estudo se torna inviável, pois analisar empiricamente todas as características de uma série, para se construir um sistema de recomendação, é humanamente inavaliável.

Nesse sentido, o estudo de Atiya (2020) demonstra que modelos de previsão combinados foram os vencedores em competições internacionais de M4. Eles afirmam que essa combinação foi possível devido ao estudo aprofundado desses modelos e à análise detalhada de suas particularidades. Em concordância, Bandara, Bergmeir e Smyl (2020)

propõem um modelo de clusterização de séries temporais baseado em redes neurais de memória longa, cujo objetivo é encontrar similaridades entre diferentes tipos de STUs para atuar em conjunto com métodos de previsão de séries temporais. Os resultados indicam que as redes neurais artificiais (RNAs) podem substituir processos que antes eram realizados por métodos como kMeans, DBScan, *Partition Around Medoids* (PAM) e Snob. Ou seja, as RNAs podem ser utilizadas diretamente para extrair e avaliar características e recomendar métodos de previsão para classes específicas de STUs.

De acordo com Asadi e Regan (2019), as redes neurais convolucionais são mais eficazes do que os métodos mais avançados de imputação de dados faltosos, pois exploram mais profundamente os dados e extraem mais características e correlações complexas entre eles. Além disso, a escolha do método a ser utilizado pode ser difícil, uma vez que diferentes resultados podem ser obtidos ao comparar as variações dos métodos, como demonstrado nos resultados apresentados por Flores, Tito e Silva (2019), onde os k-vizinhos com $k > 2$ superaram aqueles com $k \leq 2$.

De outra forma, Shah e Dimitrov (2022) apresentam um comparativo de outras técnicas de previsão com um modelo de regressão aditiva de tendências não lineares ajustadas à sazonalidade anual, semanal e diária, além dos efeitos de feriados, por meio da série de Fourier, esse modelo que denomina-se *Prophet*. O comparativo foi realizado com outros modelos com média móvel integrada autorregressivo (ARIMA) e média móvel integrada autorregressivo sazonal (SARIMA). Eles concluíram que redes neurais recorrentes de memória Longa (LSTM), o ARIMA, SARIMA, *Prophet*, Holt-Winters (HW) multiplicativo e HW aditivo, têm despenho na imputação diretamente relacionado às características da série e não aos métodos.

Em outra linha, há abordagens de imputações de dados indiretas, essas imputações são realizadas por meio de meta-aprendizado. O meta-aprendizado engloba qualquer forma de aprendizado baseado em dados adquiridos de experiências anteriores (VANSCHOREN, 2019). De acordo com Faceli et al. (2011, p.271), o meta-aprendizado inicia-se com a produção de um conjunto de dados e uma coleção de algoritmos. Dessa forma, para cada conjunto, são extraídas as principais características e anotado o desempenho dos métodos quando aplicadas a imputações e, assim, constrói-se o metadado. Posteriormente, o conjunto de metadados é utilizado em um sistema de recomendação que indique o melhor método de imputação.

De forma similar, Wu et al. (2020) propuseram uma estrutura de imputação de séries temporais sensoriais (PTSI), essa recebe uma coleção de séries temporais que passam por um *encoder-decoder* e uma camada de adaptação de parâmetros, entre essas duas camadas, após o *encoder-decoder*, há concatenação da própria saída (meta *update*) que na primeira iteração recebe parâmetros aleatórios. Por fim, a informação de meta-aprendizado é concatenada a saída da série destino, na qual os valores são imputados usando a meta-informação. Os resultados superaram métodos eficientes como GRU-D e ExGANs, os autores mostram que o meta-aprendizado é promissor e que os *encoders* utilizados em CNNs podem ser uma grande ferramenta no processo de refinamento de características para imputação por meio do meta-aprendizado, contudo não há garantias de um bom desempenho utilizando outros tipos de séries temporais.

Em um processo semelhante de imputação, Yang e Zhang (2022) propuseram uma rede híbrida composta por CNN, LSTM e meta-aprendizado para prever níveis de água no subsolo. Os dados passam por um bloco com três tipos de camadas, uma CNN que produz mapas de características e passam para uma segunda camada composta de LSTM, que capta melhor as relações de longo prazo entre as variáveis, visto que é uma rede de memória de longo prazo e, por fim, as informações são passadas para uma camada totalmente conectada para gerar o resultado da previsão.

A rede de Yang e Zhang (2022) é composta por dois blocos, o primeiro gera os metadados que são concatenados a entrada de um segundo bloco, onde é realizado o teste. Yang e Zhang (2022) Verifica que as redes já utilizadas para previsão e imputação de dados em séries temporais apresentam melhores desempenhos quando utilizadas em conjunto com meta-aprendizado. Podemos confirmar esse desempenho nos resultados mostrados, por superarem as redes LSTM e MLP, além de métodos consagrados como ARIMA, LSTMnet e regressão múltipla. Contudo, também limita-se a um tipo série UTS. A Tabela 1 mostra um resumos dos trabalhos apresentados com suas principais contribuições e características.

Tabela 1 – Resumo dos trabalhos apresentados

AUTOR (ANO)	TÉCNICAS	CONTEXTO	QTD. DE SÉRIES
Bokde et al. (2018)	imputePSF	Imputação	3
Asadi e Regan (2019)	CNN-BiLSTM	Imputação	1
Flores, Tito e Silva (2019)	K-NN	Imputação	1
Khayati et al. (2020)	Interpolação, SVDImpute, SoftImpute, DynaMMo	Imputação	8
Afrifa-Yamoah et al. (2020)	ARIMA, Suavização de Kalman, Regressão múltipla	Imputação	1
Wu et al. (2020)	PTSI ,GRU-D,SimpleFill	Imputação/ meta-aprendizado	2
Atiya (2020)	<i>Survey</i>	Previsão	–
Bandara, Bergmeir e Smyl (2020)	RNN ¹ , LSTM ²	Previsão/Clusterização	2
Shah e Dimitrov (2022)	Holt-Winters, ARIMA ,LSTM	Previsão	32
Yang e Zhang (2022)	CNN-LSTM , CNN-LSTM-ML	Previsão,Meta-aprendizagem	1

No entanto, os estudos encontrados que abordam imputação e classificação de séries temporais com auxílio de meta-aprendizado tendem a ser substitutos de outros métodos de imputação, sem garantia de que essas abordagens sejam superiores aos métodos tradicionais em todas as situações. Esse fato é corroborado por (FACELI et al., 2011), que aponta a falta de uma solução única e definitiva para imputação de dados faltosos em séries temporais.

Assim, esta pesquisa está embasada na premissa de que em uma série temporal podem existir subséries com comportamentos distintos. Para lidar com essa questão, propõe-se a subdivisão da série em partes menores, permitindo a aplicação de diferentes métodos de imputação em cada uma delas, visando otimizar os resultados da imputação. Essa estratégia é viabilizada pelo uso do meta-aprendizado.

2.1 Considerações Finais

Esta seção descreve uma revisão de estudos que sustentam a pesquisa proposta. Inicialmente, foram apresentados trabalhos que realizaram comparações entre técnicas de imputação, evidenciando a grande dificuldade na escolha da melhor abordagem para tratar dados faltantes, a fim de otimizar e aprimorar o processo. Neste estudo, busca-se superar esse problema por meio da utilização de um conjunto de técnicas de imputação, em vez de se restringir a uma única abordagem. A proposta é adotar uma abordagem não exclusiva.

Além disso, evidenciou-se que pesquisas recentes têm demonstrado a superioridade das Redes Neurais (RNs) em relação a técnicas clássicas de imputação, previsão e clusterização em séries temporais. Isso reforça o fato de que as RNs são indicadas

para abordagens inovadoras no contexto das STs. Assim, é necessário empregar técnicas baseadas em aprendizado de máquina para a imputação de dados.

Por fim, alguns estudos mostraram que é possível realizar tanto imputação quanto previsão em STs, utilizando Redes Neurais Convolucionais (CNN). Essa constatação justifica a abordagem de meta-aprendizagem e o uso de CNN neste trabalho.

3 FUNDAMENTOS TEÓRICOS

Este capítulo descreve os fundamentos teóricos necessários para o entendimento do problema abordado neste trabalho e as técnicas que embasam o método desenvolvido. São descritas a conceituação de séries temporais e sua importância; as técnicas de imputação de dados utilizadas, fundamental para o entendimento do domínio do problema; o meta-aprendizado; por fim, os conceitos e técnicas de Redes Neurais aplicadas na metodologia, como Redes Neurais *Long Short Term Memory* (LSTM) e a rede utilizada para o meta-aprendizado.

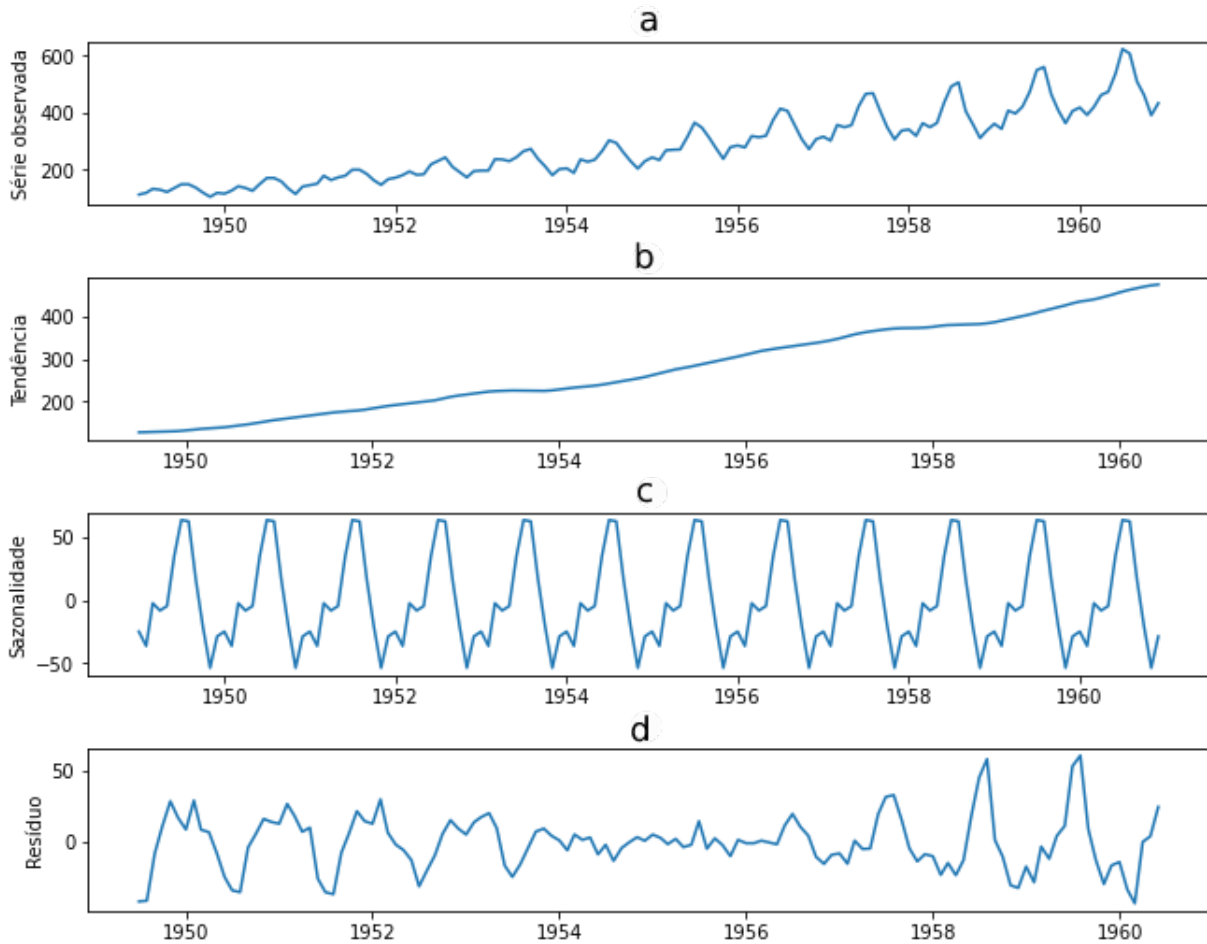
3.1 Séries Temporais

Séries temporais são uma coleção de observações realizadas sequencialmente ao longo de intervalos regulares de tempo (BROCKWELL; DAVIS, 2002). Essas observações têm como característica a dependência entre observações vizinhas. O objetivo da análise de séries temporais é identificar padrões não aleatórios na série temporal de uma variável de interesse, e a observação deste comportamento passado pode permitir fazer previsões sobre o futuro, orientando a tomada de decisões (REIS, 2015). Para representação de séries temporais, geralmente utiliza-se uma letra, por exemplo, x , e para representar-se uma medição anexa-se um valor inteiro à letra para indicar qual medição está sendo representada. Logo, x_0 seria a primeira medição da série temporal x .

Uma série temporal é composta por quatro principais componentes: Tendência, Ciclo, Sazonalidade e Ruído (HARVEY; SHEPHARD, 1993). Tendência se refere a séries monótonas que sempre cresce ou decresce, podendo ainda haver séries estacionárias quando suas componentes estatísticas como média e variância são constantes ao longo do tempo em relação a uma zona de probabilidade. A tendência pode ser entendido como a direção geral da série temporal. Vê-se um exemplo da série *Air Passengers* indicado na Figura 1(B). Geralmente as variações cíclicas e irregulares são avaliadas em conjunto. Reis (2015, p.21) diz que variações sazonais são oscilações de curto prazo, que ocorrem sempre dentro do ano, e que se repetem sistematicamente ano após ano. A Figura 1(C) demonstra a sazonalidade. Ainda segundo Reis (2015, p.26), variações cíclicas são padrões de longo prazo (superiores a um ano). Eles podem ser interpretados como as variações regulares em torno da tendência. Na Figura 1(B) que se repetem em torno de 4 anos. Por fim, o

ruído, ou resíduo (*Resid*), são os elementos que não possuem autocorreção, ou seja, não podem ser relacionados a nenhum dos outros componentes (Figura 1(D)).

Figura 1 – Componentes de uma série temporal extraídos da série *Air Passengers*.



Fonte: Do autor.

Reis (2015) diz que a formação de uma série temporal no modelo clássico pode ser feita por duas funções matemáticas, ou seja, é possível definir duas funções que geram as séries temporais. Estas funções, são nomeadas funções de formação. Essas funções são o modelo aditivo e modelo multiplicativo. A função de formação que gera uma série temporal pode possuir 2 componentes, o determinístico e um estocástico. Assim, dada uma variável objetivo y de um processo qualquer, pode-se definir as funções formadoras no modelo aditivo representada na Equação 3.1 e no modelo multiplicativo da Equação 3.2.

$$y_t = T + S + R \quad (3.1)$$

$$y_t = T \cdot S \cdot R, \quad (3.2)$$

onde T é a tendência, S sazonalidade e R o componente aleatório ou resíduos.

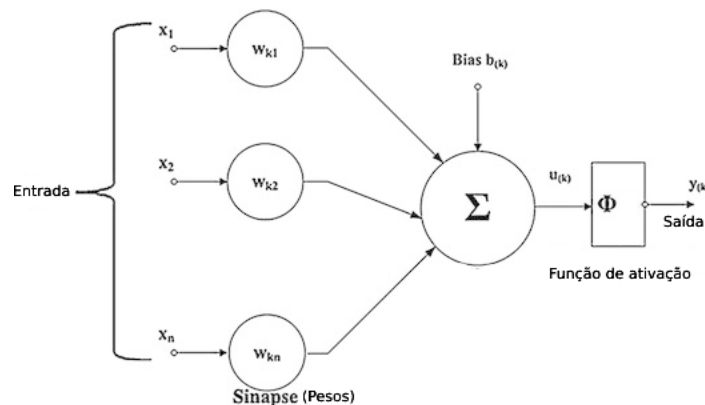
O componente estocástico (ruído), é um componente aleatório gerado por um processo estocástico. Nota-se que uma série temporal pode ter o componente determinístico, mas devido à existência de um componente estocástico se impossibilita determinar o valor exato para a função devido à existência de um elemento aleatório.

Uma série temporal pode ser contínua quando as observações são efetuadas continuamente no tempo. Assim, será denotada por $\{X(t) : t \in T\}$, $T = \{t : t_1 < t < t_2\}$, ou discreta se as observações são feitas em tempos específicos, geralmente equi-espçados. Neste caso será denotada por $\{X(t) : t \in T\}$, $T = \{t_1, \dots, t_n\}$ (EHLERS, 2021).

3.2 Redes Neurais

Baseado no funcionamento de um neurônio biológico, McCulloch e Pitts (1943) propuseram o neurônio matemático mais aceito à época: um neurônio artificial que calcula a entrada de um sinal, o qual este é ampliado, comprimido ou invertido e depois é normalizado e passado adiante, seja para outro processo ou para um resultado final.

Figura 2 – Estrutura de funcionamento de um neurônio artificial.



Fonte: Adaptado de (HAYKIN, 2010).

A Figura 2 ilustra a estrutura de um neurônio artificial, que, de acordo com Haykin (2010), apresenta os seguintes componentes básicos:

- Um conjunto de sinapses, em que cada uma possui um valor de peso, que representa a “força” da conexão. No neurônio k , um sinal de entrada x_i é multiplicado pelo peso sináptico w_{ki} , onde i varia entre $[1, m]$;
- Uma combinação linear Σ que realiza a soma ponderada dos sinais de entrada x_i conforme os respectivos pesos sinápticos w_{ki} ;
- Uma variável *bias* b_k que serve para aumentar o grau de liberdade na tendência de ajuste dos pesos;
- Uma função de ativação Φ que mapeia a solução de $\mathbb{R}^t \rightarrow \mathbb{R}^q$ com $t, q \in \mathbb{N}$ e com $t > q$.

Matematicamente temos em multiplicação matricial a seguinte representação:

$$\begin{bmatrix} 1 & x_{11} & \dots & x_{1m} \\ 1 & x_{21} & \dots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \dots & x_{nm} \end{bmatrix} X \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_m \end{bmatrix},$$

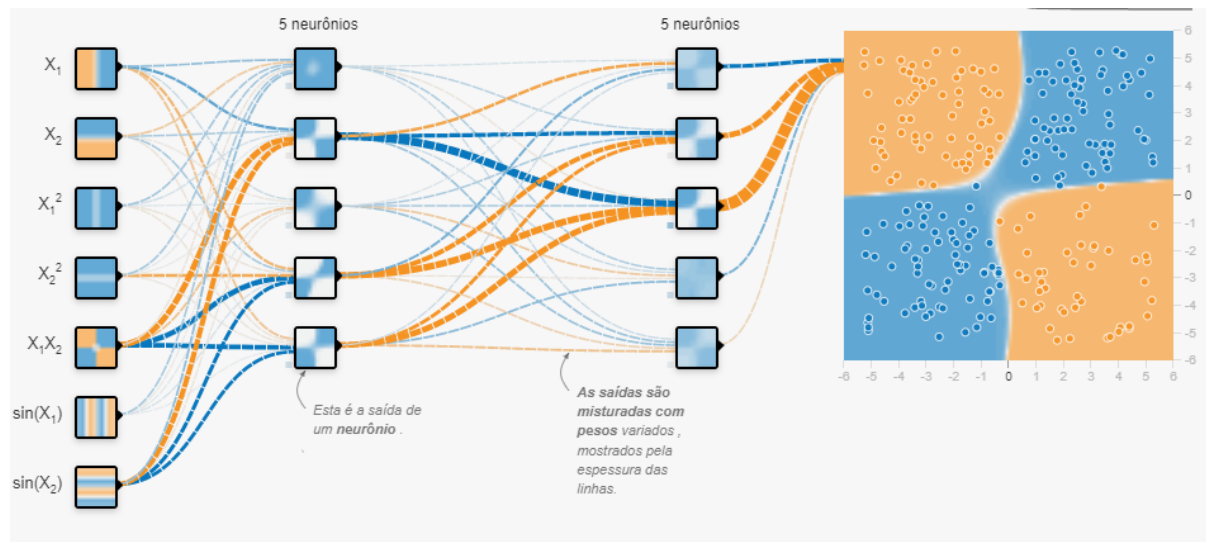
onde a matriz de entrada, ou seja, entrada da rede é multiplicada por um vetor de pesos. Assim, dessa combinação linear a rede produz os elementos y_i , $i \in [0, m]$, em um vetor resultante, esses passam por uma função de ativação, onde o usuário/programador determina um limiar para a ativação. Usualmente. Após a aplicação do limiar, as saídas poderão ser 0, para desativado, e 1, para ativado. As redes com essa configuração são muito eficientes para problemas específicos, mas deixa a desejar em problemas que requerem alto grau de generalização (FALCÃO et al., 2013).

3.2.1 Redes de Múltiplas Camadas e *Back Propagation*

Para contornar o problema da generalização fraca da rede perceptron foi proposta rede perceptron de múltiplas camadas (MLP, do inglês, *multilayer perceptron*). Além do processo de propagação direta de dados, conhecido como *feed-forward*, as MLPs também utilizam o processo de propagação reversa, chamado de *feed-backward*, no qual o erro é propagado da camada de saída para a camada de entrada. O processo *feed-forward* e *feed-backward* é conhecido como algoritmo de retro-propagação (em inglês, *Backpropagation*). A ideia principal das redes MLPs é o fato de neurônios biológicos

trocarem e compartilham informações entre si, enquanto a retro-propagação do erro está na correção do aprendizado por tentativa e erro Rumelhart, Hinton e Williams (1986). A Figura 3 exemplifica visualmente como é a estrutura de uma rede de múltiplas camadas.

Figura 3 – Arquitetura de uma rede MLP com uma camada oculta.



Fonte: <https://playground.tensorflow.org/>.

As redes MLP usam a regra de aprendizado por reforço através da correção de erro, que ocorre por meio do algoritmo de retro-propagação. O processo de aprendizagem por retro-propagação de erro consiste nos dois passos *feed-backward* e *feed-forward*. Os pesos sinápticos permanecem fixos durante a etapa de *feed-forward*, mas na retropropagação, todos os pesos são ajustados de acordo com uma regra de correção de erro, que se baseia na saída correta esperada (HAYKIN, 2004). Na Figura 3, rede MLP recebe 7 entradas derivadas de X_1 e X_2 e aplica sobre elas filtros seguidos (Matrizes de pesos), o primeiro conjunto de filtros são denominados camada escondida (primeiros 5 neurônios) e o segundo conjunto de camada de saída (últimos 5 neurônios) sendo ativados usando a função de ativação binária sigmoide (MARREIROS et al., 2008) classificam as amostras de entrada em dois grupos. Pode-se notar que cada neurônio é um classificador linear e a combinação desses gera uma classificação multilinear. Ainda se nota que a rede consegue identificar quais características são mais importantes, dando a elas maior peso, representado pela linha de espessura maior das linhas das sinapses.

O treinamento da rede MLP consiste basicamente de 12 passos, e são:

1- Inicializar os pesos e bias Seja H o número de neurônios da camada escondida, I o número de neurônios da camada de entrada e O da camada de saída, montam-

se as matrizes $W_{hi} = M_{H \times I}$, $W_{oh} = M_{O \times I}$, $bias_{hi} = M_{H \times 1}$ e $bias_{oh} = M_{O \times 1}$, onde W_{hi} , W_{oh} , $bias_{hi}$ e $bias_{oh}$ são respectivamente os pesos entre camada escondida e de entrada, pesos entre camada escondida e de saída, bias entre camada escondida e de entrada e, bias entre camada escondida e de saída. Todos são matrizes de valores aleatórios com distribuição equidistante, serão o primeiro palpite que a rede dará para a estratificação das características exploradas.

2- Calcular a entrada da camada escondida Dada as variáveis de entrada u e x a dimensionalidade de u , a entrada da camada escondida é dada por $input_h = w_{hi} \cdot u + bias_{hi} \cdot M_{1 \times x}$, onde $M_{1 \times x}$ é uma matriz unitária.

3- Calcular a saída da camada escondida Dada a função de ativação f , a saída da camada escondida é dada por $Y_h = f(input_h)$.

4- Calcular a entrada da camada de saída Seja yh a dimensionalidade de Y_h , a entrada da camada de saída é dada por $input_o = w_{oh} \cdot Y_h + bias_{oh} \cdot M_{1 \times yh}$, onde $M_{1 \times yh}$ é uma matriz unitária.

5- Calcular a saída da rede neural Dado k real a saída da rede neural é obtida por $Y = k \cdot input_o$.

6- Calcular o erro de saída O erro de saída é dado por $E = D - Y$, onde D são os valores esperados.

7- Calcular a variação dos pesos entre camada de saída e escondida Dado o neurônio i de saída e o neurônio j escondido, a variação entre i e j é calculada pela seguinte equação $\Delta w_{ij} = \eta \cdot e_i \cdot f'_i \cdot out_j$, onde η é uma constante denominada taxa de aprendizado, e_i o erro do neurônio i , f'_i a derivada da função de ativação do neurônio i e out_j a saída do neurônio escondido j .

Assim, pode-se realizar um só cálculo. Dada as dimensões de $input_o = t$, calcula-se a derivada da função de ativação da camada de saída $df = k \cdot M_t$, M_t é uma matriz unitária. Disso decorre que para se corrigir o bias tem-se $\Delta bias_{oh} = \eta \cdot \left[\sum (E \otimes df)^T \right]^T$ e $\Delta W_{oh} = \eta \cdot (E \otimes df) \cdot Y_h^T$, onde \otimes é o produto Hadamard (MESQUITA et al., 2014).

8- Calcular o erro retro-propagado Dado o neurônio i de saída e o neurônio j escondido, o erro retro-propagado é dado por $e_j = - \sum_1^n e_i \cdot df'_i \cdot w_{ij}$, onde e_j é o

erro do neurônio escondido j , e_i o erro do neurônio de saída i e w_{ij} o peso entre esses dois neurônios. Da mesma forma, pode calcular o erro em um só cálculo $E_h = -W_{oh} \cdot (E \otimes df)$.

9- Calcular a variação dos pesos entre camada escondida e de entrada Dado o neurônio i de saída e o neurônio j escondido, a variação entre i e j é calculada pela seguinte equação $\Delta w_{jk} = -\eta \cdot e_j \cdot f'_j \cdot u_k$, onde η é uma constante denominada taxa de aprendizado, e_i o erro do neurônio i , f'_j a derivada da função de ativação do neurônio i e u_k o neurônio k de entrada. O valor negativo é posto para se descolar no sentido contrário apontado pelo gradiente, percorrendo assim para o menor valor possível obtido pela função de custo, ou seja, de avaliação do erro global.

Para se realizar um só cálculo, pode-se obter a variação dos bias por $\Delta bias_{hi} = -\eta \cdot \left[\sum (E_h \otimes f')^T \right]^T$ e a variação dos pesos dado por $\Delta W_{hi} = -\eta \cdot (E_h \otimes f') \cdot u^T$.

10- Atualizar os valores dos pesos e bias Os valores dos pesos são atualizados utilizando $W_{hi} = W_{hi} + \Delta W_{hi}$, $W_{oh} = W_{oh} + \Delta W_{oh}$, $bias_{oh} = bias_{oh} + \Delta bias_{oh}$ e $bias_{hi} = bias_{hi} + \Delta bias_{hi}$.

11- Calcular o erro global A função de custo mais utilizada para se calcular o desempenho geral da rede é o MSE (da sigla em *inglês* *Mean Squared Error*) dado por $MSE = \frac{1}{n} \sum (D - Y)^2$, onde n é a quantidade de amostras.

12- Verificar se erro global é maior que erro máximo esperado Caso a verificação seja verdadeira retorna-se ao passo 2.

É importante destacar que tanto as funções de ativação da camada escondida e da camada de saída podem ser as mesmas. Neste trabalho descreve-se a função genérica para a camada escondida e para a camada de saída a função linear. Também pode-se utilizar bias ou não, bem como outra função de custo para o erro global. Essa rede possui diversas vantagens sobre a Perceptron, a MLP de uma camada, mas também continua a ter muitas desvantagens, a maior delas é a perda do gradiente ao passo que se adiciona novas camadas escondidas.

3.2.2 Aprendizagem Profunda

O aprendizado profundo é uma área do aprendizado de máquina cujo objetivo é isolar características elementares para que os algoritmos de aprendizagem possam construir um conhecimento abstrato utilizando relações hierárquicas entre as características dos dados observados (GUO et al., 2016).

Essas técnicas forneceram aos computadores o poder de construir seus próprios conhecimentos. Essa capacidade vem atraindo diversas áreas do conhecimento, principalmente quando se trata de processamento de grandes quantidades de dados (NAJAFABADI et al., 2015). Goodfellow, Bengio e Courville (2016) mostram que para se entender um conceito complexo é necessário conhecer a essência do conceito, ou seja, conhecer conceitos mais simples.

O aprendizado realizado utilizando diversas camadas escondidas denota-se aprendizado profundo. O aprendizado profundo é possível por meio a redundância, bem como a exploração disjunta das camadas escondidas de várias características diferentes.

Assim o aprendizado profundo é aplicado nas mais diversas áreas com diferentes fins e mostrando sempre competitividade ou superioridade (SHINDE; SHAH, 2018). Atualmente os algoritmos de aprendizagem profunda são reconhecidos pelo excelente desempenho em tarefas que envolvem processamento, reconhecimento de padrões, tratamento, regressão e clusterização de imagens e vídeos.

3.2.3 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (CNNs, do inglês *Convolutional neural network*) são hoje uma dos modelos de redes de aprendizado profundo mais conhecidos. A principal característica dessas redes são suas camadas convolucionais, que aprendem as características das entradas por meio de campos receptivos locais, ou seja, matrizes menores que comumente se denomina filtros. Esses filtros são equivalentes aos neurônios nas redes MLPs.

3.2.3.1 Camada Convolutional

De acordo com Ponti e Costa (2018), nas camadas convolucionais cada neurônio é um filtro aplicado a uma imagem de entrada e cada filtro é uma matriz de pesos. Assim, dada uma imagem RGB $28 \times 28 \times 3$, em uma rede MLP teríamos para cada sinapse

2.352 pesos, em vez disso definimos um filtro $\kappa \times \kappa \times D$, onde κ é a primeira e a segunda dimensão do filtro e D a dimensão corresponde a imagem de entrada. Escolhendo-se $\kappa = 5$, obtém-se um filtro $5 \times 5 \times 3$ que corresponde a $5 \times 5 \times 3 = 45$ pesos.

Assim, dados n filtros, esses serão aplicados a um mesmo local de uma imagem (campo receptivo) e aplica-se o somatório do produto entre esses filtros dados pela Equação 3.3.

$$(f * g)(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j)g(x - i, y - j), \quad (3.3)$$

onde (x, y) e (i, j) são coordenadas da imagem e dos filtros, respectivamente.

Esse processo é realizado movimentando os filtros sobre a imagem, caso se queira preservar o tamanho da imagem usa-se o preenchimento das bordas com zeros (*padding same*), caso contrário, usa-se *padding valid*. O movimento (*stride*) dos filtros é dado por um número natural e representa em quantos pixels o filtro será transladado para realizar a próxima operação.

Após as camadas convolucionais, comumente usa-se uma função de ativação. A função mais utilizada é a função ReLU, dada por:

$$f(x) = \max(0, x) \quad (3.4)$$

3.2.3.2 Camadas *Fully Connected* (FC)

As camadas FCs, também conhecidas como camadas totalmente conectadas, são utilizadas nas redes MLPs e, após blocos sucessivos de convolução, visam aumentar a dimensionalidade das características processadas. Isso permite a análise direta de grupos de características e a identificação da redundância produzida por diferentes convoluções. Isso é possível porque, para cada neurônio, há n dados em uma camada FC, resultando em um tensor $(K \times k \times d)$ e $k^2 \times d$ pesos. No total, a quantidade de pesos nos neurônios é de $k^2 \times d \times n$.

Antes de chegarem às camadas totalmente conectadas (FCs), é comum que os tensores sejam transformados, sendo achatados e redimensionados para uma única dimensão. Esse processo é conhecido como *Flatten*.

3.2.3.3 *Batch Normalization* (BN)

A operação de normalização de batch (BJORCK et al., 2018) pode ser explicada em quatro passos principais. Primeiro, por meio da Equação 3.5, o BN determina a média do lote. Segundo, pela Equação 3.6, determina a variância do lote. Posteriormente, normaliza o vetor de ativação com a Equação 3.7, que possibilita uma distribuição normal equidistante. Por fim, ajusta a distribuição utilizando γ e β .

$$\mu = \frac{1}{n} \sum_i Z^{(i)}, \quad (3.5)$$

onde $Z^{(i)}$ é o vetor de ativação, μ a média do lote e n a quantidade de elementos do lote.

$$\delta^2 = \frac{1}{n} \sum_i (Z_{(i)} - \mu)^2, \quad (3.6)$$

onde δ^2 é a variância do lote e n a quantidade de elementos do lote.

$$Z_{norm}^{(i)} = \frac{Z^{(i)} - \mu}{\sqrt{\delta^2 - \epsilon}}, \quad (3.7)$$

onde $Z_{norm}^{(i)}$ é o vetor de ativação normalizado, n a quantidade de elementos do lote e ϵ é uma constante usada para estabilidade numérica.

$$\hat{Z} = \gamma * Z_{norm}^{(i)} + \beta, \quad (3.8)$$

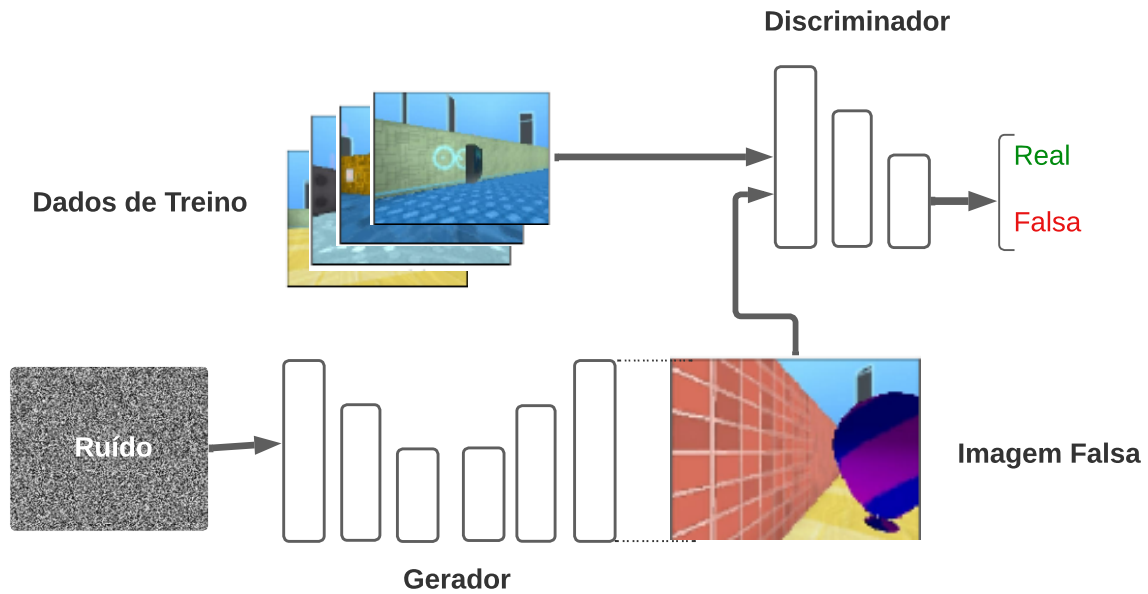
onde \hat{Z} é o lote com a distribuição ótima, σ e β dois parâmetros treináveis. γ possibilita o ajuste de δ e β o ajuste do bias.

3.2.4 *Técnica de Imputação Baseada em Aprendizado de Máquina*

Uma Rede Adversária Generativa (GAN) (TORRE, 2023) é composta por uma rede denominada Gerador que produz novas instâncias de dados enquanto e outra denominada Discriminador que decide se uma instância de entrada pertence ao conjunto de imagens reais do treino ou não. Na Figura 4 verifica-se esse processo.

Em particular tem-se a rede Pix2Pix GAN. Essa será tomada como exemplo para explicar esse tipo de rede. Dessa forma, no que diz respeito a métodos baseados em aprendizado de máquina, este estudo propôs um método de imputação de dados ausentes a partir da rede Pix2Pix GAN (ISOLA et al., 2016).

Figura 4 – Estrutura De uma Rede Adversária Generativa



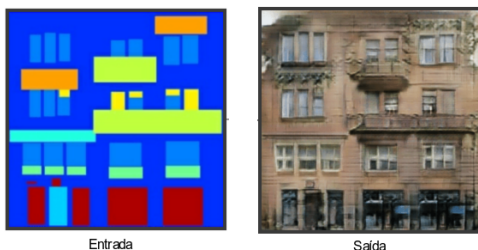
Fonte: Acervo do autor.

A rede Pix2Pix é uma rede que usa Redes Adversárias Generativas (GAN's) para traduzir imagens, ou seja, converte uma imagem em outra. Essa rede é muito útil na área de processamento de imagens, algumas de suas aplicações estão em colorir imagens artificialmente a partir de bodas e desenhos (Figura 5a) e em um processo contrário pode gerar imagens mais simples como mapas a partir de imagens de satélite Figura (5b).

Figura 5 – Tradução de usando Pix2Pix

(a) Imagem de fachada

(b) Imagem de satélite



Fonte: Adaptado de (HESSE, 2017)



Fonte: Adaptado de (ISOLA et al., 2017)

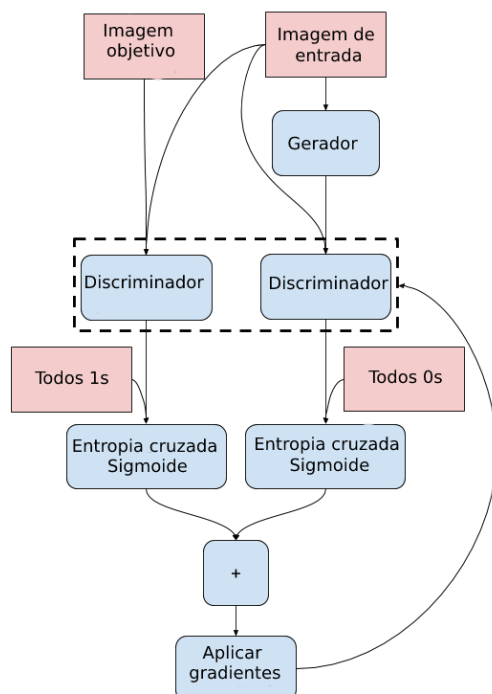
A estrutura de rede Pix2Pix é composta por duas redes, a primeira rede é uma U-Net (RONNEBERGER; FISCHER; BROX, 2015a) e um *Encoder*. A rede U-net é a rede responsável pela geração de uma imagem falsa e recebe o nome de gerador, por outro lado, o *Encoder* recebe o nome de Decodificador e tem objetivo de diferenciar uma imagem real de uma imagem falsa.

A rede em questão recebe uma imagem como entrada, o que a distingue de uma *Generative Adversarial Network* (GAN) tradicional (BAU et al., 2018), que recebe um ponto de espaço latente em vez disso, ou seja, uma informação genérica de um conjunto de dados (ARVANITIDIS; HANSEN; HAUBERG, 2017). A aleatoriedade na rede é introduzida através das camadas de *dropout*, que eliminam parte do aprendizado de uma camada da rede neural. A saída dessa rede é uma outra imagem no domínio do espaço de destino.

Por outro lado, o discriminador que é o *Encoder* recebe duas imagens de entrada, uma imagem real e a outra imagem gerada pelo gerador. O discriminador concatena as duas imagens que passam por camadas de *Downsampling* onde são re-amostradas e comprimidas e ao final tem como saída uma probabilidade de pertencer a uma das duas classes, verdadeira ou falsa. A rede Pix2Pix foi projetada para classificar não a partir da imagem inteira, mas a partir de trechos da imagem por meio do discriminador Markoviano (PatchGAN).

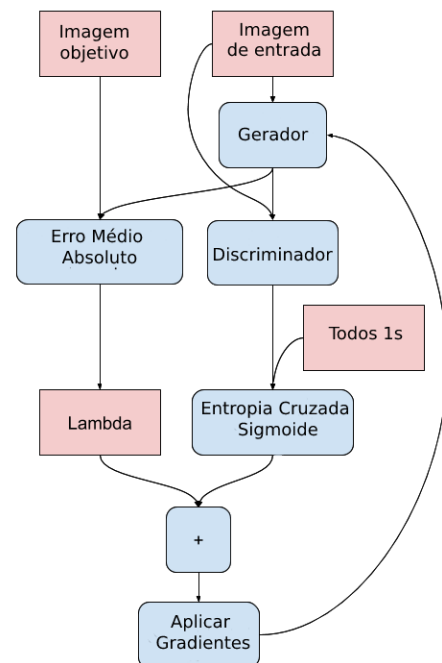
Figura 6 – Componentes da rede Pix2Pix

(a) Fluxo de treinamento para o Discriminador



Fonte: Adaptado de (ABADI et al., 2015)

(b) Fluxo de treinamento para o Gerador



Fonte: Adaptado de (ABADI et al., 2015)

A Figura 6 mostra o fluxo de treinamento da rede Pix2Pix discretizado para o Gerador e Discriminador. Pode-se ainda notar que o aprendizado da rede tem por base

o Erro Absoluto Médio (MAE), a função de ativação sigmoide e uma função de custo baseada na sigmoide.

3.3 Técnicas de Clássicas de Imputação

A etapa de processamento de dados é importante em qualquer análise, mas é comum a ocorrência de períodos sem informações disponíveis nos dados. Isto se deve a diversos fatores, como erros humanos ou falhas em equipamentos de medição. Estes dados ausentes, também conhecidos como falhas, são um dos principais problemas no estudo de séries temporais. Entre as estratégias usadas para resolver este problema, estão os métodos de preenchimento de falhas, também conhecidos como métodos de imputação de valores ausentes (JÚNIOR, 2022). Este trabalho selecionou 11 técnicas de imputação, incluindo técnicas estatísticas e baseadas em aprendizado de máquina.

As primeiras técnicas selecionados são a média e mediana. Esses métodos são muito conhecidos e amplamente utilizados em estudos pela comunidade científica, eles consistem necessariamente na substituição de valores únicos ou trecho de valores faltosos denominados (*gaps*) pela média e mediana. Essas medidas podem variar conforme a abordagem tomada. Podem ser calculadas a partir de observações de todas as amostras ou amostra específica, de um grupo que atende a certas características ou a regras especiais definidas pelo autor. Nesse trabalho, a substituição de valores faltosos pela média e mediana considerará todas as amostras disponíveis.

Adicionalmente, têm-se os métodos *Backfill* e LOCF (*Last observation carried forward*), que substituem o valor faltoso de observação pelo valor da próxima observação e valor da observação anterior, respectivamente. O método *Backfill* não substitui *gaps* quando estes estão no início da série temporal, antes de todas as observações válidas.

Outros métodos comuns no tratamento de dados são as interpolações. O método de interpolação mais simples é o linear. No qual, para dois pontos $p_1 = (x_0, y_0)$ e $p_2 = (x_1, y_1)$, calcula-se o ponto $(x, y) \in [p_1, p_2]$ pela Equação 3.9. Além disso, para o cálculo da variável dependente y , a variável independente x deve ser conhecida.

$$y = y_0 + (y_1 - y_0) \frac{x - x_0}{x_1 - x_0} \quad (3.9)$$

Outras interpolações, de grau maior que um, são melhor entendidas pela aproxi-

mação de um polinômio da forma

$$P_n(X) = a_n X^n + a_{n-1} X^{n-1} + \cdots + a_1 X + a_0. \quad (3.10)$$

No entanto, a interpolação polinomial largamente utilizada é a de Lagrange (MENEZES; DUARTE, 2010) (Equação 3.11), por ser muito mais eficiente e de menor custo.

$$P_n(X) = \sum_0^n y_i \prod_{j=0; j \neq i}^n \frac{X - X_j}{X_i - X_j}, \quad (3.11)$$

com $n \in \mathbb{N}$.

Dessa forma, para calcular a interpolação de grau n , basta utilizar a equação de interpolação de Lagrange.

Algumas interpolações são mais complexas e de alto custo computacional, como interpolação por *splines*. Este método consiste em uma função definida por partes, na qual cada parte é um polinômio de grau $n - 1$ controlado por n pontos. As B-splines são uma combinação de *splines* de mesmo grau B (WOLD, 1974). Assim, pode-se definir a *Spline univariada* na base *B-spline* segundo a equação:

$$S(t) = \sum_{j=0}^{n-1} p_j B_{j,k}(t), \quad (3.12)$$

onde $B_{j,k}$ é a j -ésima função de base da *B-spline* de grau n , k nós e p_j é um ponto de controle, calculada conforme a equação:

$$B_{j,k}(t) = \begin{cases} \begin{cases} 1 & \text{para } t_j \leq t < t_{j+1} \\ 0 & \text{Caso contrário} \end{cases} & \text{se } k = 1; \\ \begin{cases} \frac{t - t_j}{t_{j+k-1} - t_j} B_{j,k-1,t}(t) + \frac{t - t_j}{t_{j+k-1} - t_i} B_{j+1,k-1}(t); & \text{se } k > 1. \end{cases} \end{cases} \quad (3.13)$$

O valor de k determina o número de segmentos da *spline*, cada um definido por um conjunto de nós adjacentes. Portanto, aumentar o valor de k aumenta a flexibilidade da *spline*, mas também aumenta a complexidade computacional. Em geral, escolhe-se um valor de k que forneça a flexibilidade necessária para ajustar a curva aos dados, mas que também mantenha o custo computacional razoável.

Além dos métodos apresentados, utilizou-se a média móvel simples (MMS). Dado um conjunto de pontos $A = (a_1, a_2, \dots, a_n)$ e $m < n | m, n \in \mathbb{N}$, a MMS é definida por

$$MMS = \begin{cases} a_i^* = \sum_{k=0}^i a_k & \text{se } i \leq m; \\ a_i^* = \frac{1}{m} \sum_{k=0}^{m-1} a_{i-m+k} & \text{se } i > m, \end{cases} \quad (3.14)$$

onde a_i^* é um termo qualquer predito pela MMS e m o tamanho da janela utilizada. Neste trabalho utilizou-se a MMS com janelamento $m = 3$.

Por fim, para métodos estatísticos, tem-se o *K-Nearest Neighbors* (K-NN). Adotou-se o modelo de Flores, Tito e Silva (2019), o 4-NN. Esse método utiliza a média dos 4 vizinhos mais próximos e, notavelmente, não imputa valores ausente para *gaps* que vão além disso.

3.3.1 Redes *Long Short Term Memory*

De acordo com Staudemeyer e Morris (2019), as redes neurais *Long Short Term Memory* (LSTM) é uma variação da *Recurrent Neural Networks* (RNN). As duas redes contêm, além da entrada normal, uma entrada de uma iteração anterior. A Figura 7 mostra a rede em uma representação gráfica na forma reduzida, porém com o laço pode-se interpretar como uma representação estendida da rede, conforme a Figura 8.

Figura 7 – Forma resumida de uma RNN.

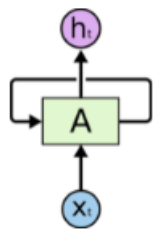
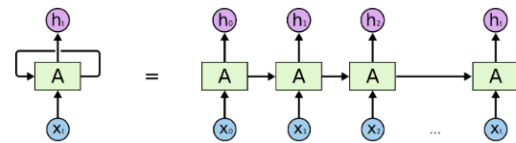


Figura 8 – Forma estendida de uma RNN.



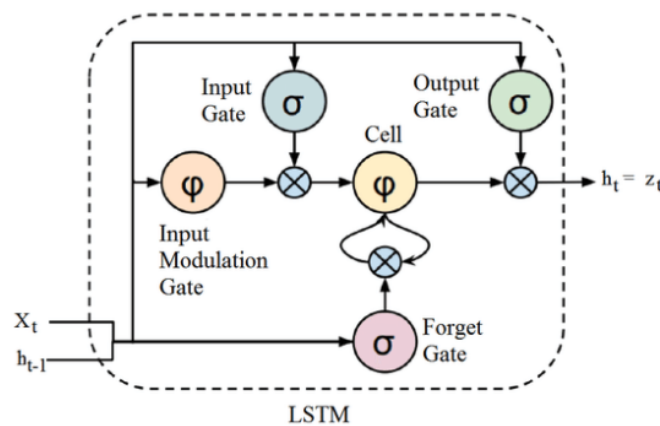
Fonte: (KANG, 2017)

A principal característica das RNNs parte da observação do aprendizado humano, que utiliza o aprendizado de experiências anteriores, principalmente recentes, para otimizar o processo de aprendizado do estado presente (KARPATHY; FEI-FEI, 2015) e, no caso de informações com correlações fortes em períodos relativamente curtos, elas funcionam muito bem. Contudo, existem casos que a correção forte está entre elementos de diferentes períodos de tempo e cria a possibilidade da rede não conseguir conectar a informação ao

ponto no qual ela é necessária. As redes LSTM foram projetadas especificamente para evitar o problema com informações de longo prazo, uma vez que essas informações podem ser essenciais para a representação correta das características dos dados.

A LSTM possui uma estrutura em cadeia composta por quatro redes neurais (Figura 9) e uma estrutura em blocos de memória denominados células. As informações são armazenadas pelas células e as atualizações de memória são feitas pelos portões (do inglês, *gates*). Existem três tipos diferentes de portões, a saber *Forget Gate*, *Input Gate* e *Output Gate*.

Figura 9 – Rede LSTM padrão



Fonte: (KANG, 2017)

O *Forget Gate* é responsável pela remoção das informações dispensáveis nas células. Esse portão recebe duas entradas: X_t (entrada no momento t) e h_{t-1} (saída de célula anterior). O resultante é ativado por uma função que fornece uma saída binária, eventualmente uma sigmoide. Essa função de ativação tentará empurrar os resultados para seus limites inferiores e superiores, ao passo que quanto mais próximo de 0, maior é a probabilidade da informação ser descartada e quanto mais próximo de 1, de ser mantida.

O *Input Gate* é responsável pela adição de informações úteis. Esse portão utiliza as mesmas entradas do *Forget Gate* que, antes de serem processadas pelo portão, passam por ativação de uma sigmoide que regula essas entradas. Em seguida, o portão ativa a entrada usando uma função tangente hiperbólica (\tanh) que tem um espaço de probabilidade maior. Assim, a saída da \tanh e a saída modulada são multiplicadas para preservar as informações que os dois processos julgaram ser úteis.

Por fim, tem-se o *Output Gate*, com a tarefa de extrair informações úteis do estado da célula atual. Primeiro, aplica-se a função \tanh nas informações que chegam a célula.

Então, a informação é regulada pela sigmoide que filtra os valores a serem lembrados usando as entradas h_{t-1} e X_t . Os valores da saída da célula e valores regulados são multiplicados para preservar as informações úteis por meio da ampliação da probabilidade resulte da multiplicação para serem enviados como uma saída e entrada para a próxima célula.

A memória de longo prazo comumente se denomina estado da célula. As setas em loop indicam o processo recursivo desta. Por meio dessa recursão, as informações dos intervalos anteriores são armazenadas na célula LSTM. Por esse motivo, essa rede é muito aplicada em processos de modelagem de linguagem, tradução de idiomas, legendas em imagens, geração de texto e *chatbots* agora ganharam espaço em diversas aplicações relacionadas à séries temporais.

3.4 Meta-aprendizado

Um grande desafio no aprendizado de máquina é tornar o aprendizado contínuo, ou seja, não ter sempre que iniciar o aprendizado do zero. Para contornar esse problema, diversos estudos foram realizados com objetivo de se otimizar o próprio algoritmo de aprendizado usando aprendizado anterior de outros algoritmos (HOSPEDALES et al., 2021).

O processo de aprendizado supervisionado de máquina convencional geralmente parte de um conjunto $X = \{(x_1, y_1), \dots, (x_n, y_n)\}$, onde x_n é a n -ésima entrada da rede e y_n o n -ésimo rótulo ou valor desejado para a entrada x_n . Nesse caso, o aprendizado é realizado fazendo o treinamento para o modelo $\hat{y} = g_\theta(x)$, parametrizando θ e resolvendo a equação:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \zeta(X, \theta, \lambda), \quad (3.15)$$

onde ζ é a função de perda que mede o erro entre a classe ou valor verdadeiro e a classe ou valor predito pelo modelo g_θ ; e λ são dados do aprendizado generalizado, geralmente pesos, que mostram como a rede deve decidir para prever o resultado. Esse modelo é avaliado em outro conjunto de dados de validação, que não entram no conjunto de dados de treinamento, no qual o rótulo/valores reais são conhecidos. Geralmente há um terceiro conjunto, denotado conjunto de teste, cujos dados não estão nos conjuntos citados anteriormente. Esse último conjunto permite avaliar a generalidade do aprendizado para

futuros casos que estão no mesmo escopo dos dados de treinamento, mas assumindo-se que são dados que serão coletados futuramente.

Diferente deste processo, o meta-aprendizado é realizado construindo-se inicialmente o metadado. O metadado é construído aplicando-se, por exemplo, um conjunto de algoritmos de aprendizado convencional onde são anotados os resultados de cada algoritmo. Esses resultados são passados para um algoritmo de aprendizado, visando aprender como os algoritmos anteriores aprenderam e, a partir daí, aprender com os resultados armazenados no metadado. Assim, dado o desempenho λ em um conjunto de tarefas (metadado) $\psi(\Gamma)$, onde Γ é um resultado específico para uma função de perda em um conjunto de dados, $\Gamma = \{X, \zeta\}$. Formalmente definimos o processo de meta-aprendizado pela equação:

$$\min_{\lambda} \mathbb{E}_{\Gamma} \approx \psi(\Gamma) \zeta(X\theta, \lambda), \quad (3.16)$$

onde ζ é a função de perda regular do aprendizado da construção do metadado e X o conjunto de dados treino e validação. Então, define-se a nova função perda por

$$\zeta(X, \lambda) = \zeta(X^{val}, \theta^*(X^{train}, \lambda), \lambda), \quad (3.17)$$

onde θ^* são os parâmetros obtidos pelo processo de construção do metadado.

3.5 Métricas de Avaliação

Os experimentos realizados nesse estudo optou-se pelo uso do Erro Médio Absoluto (MAE), que é menos sensível a desvios em comparação com o Erro Médio Quadrático (RMSE), e, portanto, mais adequado para lidar com variações na série. O RMSE fornece uma visão geral do erro. Em contrapartida, o MAE é uma das métricas usadas para avaliar a diferença absoluta entre as imputações previstas pelo método avaliado e o valor esperado, sem se sensibilizar à variabilidade dos dados. Ainda que o RMSE forneça uma visão geral, o *Symmetric Mean Absolute Percentage Error* (SMAPE), descreveremos a seguir, também o fornece essa visão, por isso não utilizaremos o RMSE.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.18)$$

Onde y_i é o valor real e \hat{y}_i o valor predito.

Para uma avaliação mais global e percentual, propôs-se a métrica *Adapted Symmetric Mean Absolute Percentage Error* (ASMAPE) que consiste em uma adaptação do SMAPE (KREINOVICH; NGUYEN; OUNCHAROEN, 2014) convencional para resolver os casos de indeterminação. O SMAPE e ASMAPE (3.19) são semelhantes, pois a primeira desconsidera as indeterminações, enquanto a segunda apenas as altera para zero. Decidiu-se evidenciar tal abordagem, pois na literatura atual tais abordagens não são explicitadas ao leitor.

$$ASMAPE = \frac{1}{n} \sum_{i=1}^n \left\{ \begin{array}{ll} 0, & \text{se } y_i = \hat{y}_i = 0 \\ \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}, & \text{caso contrário} \end{array} \right\} \quad (3.19)$$

onde y_i é o valor real, \hat{y}_i o valor predito e n a quantidade de amostras.

4 MATERIAIS E MÉTODO

Este capítulo descreve a base de dados, o método de imputações proposto que é baseado em meta-aprendizado, combinando 11 técnicas distintas de imputação, sendo uma delas uma técnica proposta baseada na rede Pix2Pix GAN. Inicialmente, apresenta-se a base de séries temporais na qual o método desenvolvido foi aplicado, seguida pela preparação das séries para imputação por utilizando todas as técnicas disponíveis. As técnicas de imputação foram aplicadas para se obter o desempenho de cada uma, com a construção do metadado como etapa preliminar. Além disso, será detalhada a rede proposta para o meta-aprendizado, e serão fornecidas informações sobre o hardware e software utilizados no desenvolvimento desta pesquisa.

4.1 Aquisição da Base de Dados

A base de dados foi construída a partir de bases de dados públicas. Os critérios de busca de dados foram, *datasets* públicos de séries temporais univariadas. Para esse estudo selecionaram-se séries temporais numéricas com dados completos, ou seja, sem valores faltosos, por ser necessário ter os dados originais para que se possa avaliar o erro real das imputações.

Foram escolhidos 15 *datasets* provenientes de 12 áreas distintas, incluindo, por exemplo, energia, número de passageiros aéreos, compartilhamento de veículos, produção de bebidas e pandemia. Ao todo, foram obtidas 777 séries com uma variedade de padrões de tendência, sazonalidade, ciclos e estacionariedade.

4.1.1 *Datasets* com Tendência e Sazonalidade

O primeiro *dataset* que apresenta tendência é o *Air Passengers* (AP) (BROWNLEE, 2019) (Figura 10a). Com dados de passagens aéreas mensais, o dataset possui 144 amostras dos anos de 1949 a 1960 e possui apenas uma série com valores variando de 104 a 622. A Figura 10a mostra que seus ciclos se repetem por períodos entre 1 e 1,5 anos.

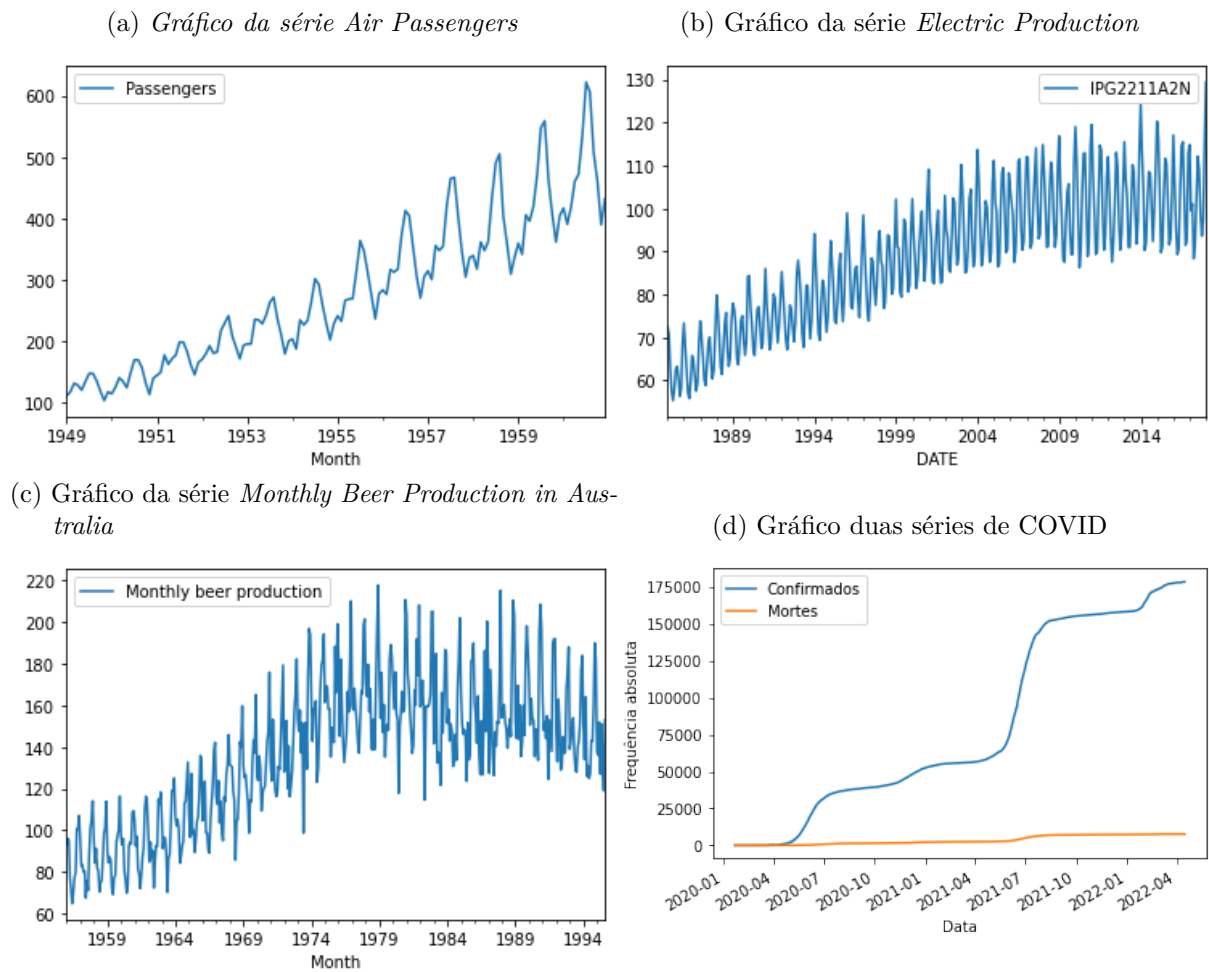
Em uma área diferente, adquiriu-se o dataset *Electric Production* (EP) (US, 2022) (Figura 10b), que contém dados da produção industrial de serviços de utilidade pública, elétricas e de gás, nos Estados Unidos, dos anos 1985 a 2018. Os dados possuem anotação diária com mínimo de 55,3151 e máximo de 119488. Na Figura 10b nota-se que a

sazonalidade se mostra com certa regularidade dentro de 1 ano. No entanto, a visualização dos ciclos não são triviais.

O terceiro *dataset* é o *Monthly beer production in Australian* (MBPA) (BAHNSEN, 2021) (Figura 10c), que contém dados mensais da produção de cerveja na Austrália. Os dados no período de 1956 a 1955 tem valor mínimo de 64,8 e máximo de 217,8. Na Figura 10c observa-se que a série apresenta comportamento semelhante ao *Electric Production*, mas com sazonalidade com mais oscilações.

Por fim, tem-se o *dataset Covid-19* (Figura 10d, contendo dados diários da Covid-19 no período de 2020 a 2021. Possui cinco principais variáveis: (*Country/Region, Province/State, Confirmed, Recovered, Deaths*), mas apenas *Confirmed, Country/Region* e *Deaths* foram selecionadas por não apresentarem valores faltosos. Assim, foi efetuada a divisão considerando as sub-variáveis países, gerando 394 séries únicas de casos confirmados e de óbitos com valor mínimo de 0 e máximo de 80625120.

Figura 10 – Gráfico das séries com tendência



Fonte: Acervo do autor.

A Tabela 2 mostra a quantidade de séries univariadas que o *dataset* possui, a fonte onde pode ser encontrada e o intervalo de valores das séries.

Tabela 2 – Principais características dos *datasets* das séries com tendência

Dataset	Quantidade de séries	Fonte	Intervalo de valores
<i>Air Passengers</i>	1	(BROWNLEE, 2019)	[104 , 622]
<i>Electric Production</i>	1	(US, 2022)	[55,31 , 129,40]
<i>Monthly beer production in Australian</i>	1	(BAHNSEN, 2021)	[64,8 , 217,8]
<i>Covid-19</i>	394	(DATA, 2022)	[0, 8E+7]

4.1.2 Datasets com Séries Estacionárias

As séries estacionárias são séries que apresentam aleatoriedade em torno de uma média constante (MORETTIN; TOLOI, 2018). Há duas principais abordagens para identificar se uma série é estacionária. A primeira é uma abordagem visual onde se traça a média da série e depois verifica se há simetria na série temporal. Contudo, a segunda

abordagem é mais confiável, o teste *Augmented Dickey Fuller* (ADF test). O ADF teste verifica a probabilidade de existir raiz unitária em uma série temporal (MACKINNON, 2010). Para isso, ele toma, respectivamente, Hipótese Nula (H0) e Hipótese Alternativa (H1) como: “A série temporal não é estacionária”, “A série temporal é estacionária”.

No teste estatístico é avaliado o valor do ADF, o *P-value* e os valores críticos. Caso o valor ADF seja menor que os valores críticos e *p-value* seja menor que um nível de significância adotado, dizemos que com nível de significância adotado podemos negar H0 e aceitamos H1, caso contrário afirmamos H0. A seguir, apresentam-se os primeiros três exemplos desse processo.

O primeiro *dataset* com série estacionária é o *Aeration rate* (AR) (DUNN, 2018) (Figura 11a), que com significância de 5% apresentou ADF = -12,31, *p-value*=0,00 e valores críticos de 1% = -3,44, 5% = -2,86 e 10% = -2,56. *Aeration rate* são dados de fluxo total de ar adicionado a um tanque de aeração¹, em litros, com 573 observações durante 1 minuto. Os tanques de aeração dependem da quantidade de ar adequado para manter partículas sólidas em suspensão, de forma que, se for adicionado muito ar, há a perda de produto sólido valioso.

Com significância de 5%, ADF = -5,52, *p-value*: = 0,00 e valores críticos de 1% = -3,435, 5% = -2,86 e 10% = -2,56 temos o *dataset Ammonia*(AMM) (DUNN, 2018) (Figura 11b), esse marca a quantidade de amônia que uma corrente de água possui em uma estação de tratamento de água. Os dados são de medições realizadas a cada 6 horas. Esse *dataset* possui 1,440 amostras sintéticas, mas que simulam dados reais.

O *dataset* de dados de vendas de bebida, *Beer sales* (BS), (Figura 11c), contem 192 amostras. Em apenas uma série não há indicação de onde são as vendas, tão pouco do período das vendas. Os dados de vendas variam entre 9,8413 a 17081. *Beer sales* (GRAVES, 2022) obteve, a 5% de significância, ADF = -12,79 *P-value* = 0,00 e valores críticos de 1% = -3,46, 5% = -2,87 e 10% = -2,57.

O *dataset Bike sharing* (BSH) (FANAEE-T; GAMA, 2013) (Figura 11d) possui duas principais séries, uma com amostras horárias e outra com amostras diárias de aluguel de bicicletas realizados e devolvidos em locais diferentes. A série adotada de *Bike sharing* foi a série de dados horários, e nessa, apenas a subséries “cnt”, com 17389 registros do total de bicicletas alugadas nesse período. Os aluguéis variam de 1 a 977 por do ano

¹ Aeração é uma etapa do tratamento de água cujo objetivo é torná-la adequada para consumo humano ou para uso em processos industriais (FURQUIM, 1978).

de 2011 a 2012. *Bike sharing* possui duas principais séries, uma com amostras horárias e outra com amostras diárias de aluguel de bicicletas realizados e devolvidos em locais diferentes.

O *dataset Sunspots* (SU) (DATA/IMAGE; BELGIUM; BRUSSELS, 2021)(Figura 11e) contém dados do número internacional de manchas solares diárias do ano de 1818 a 2022. O número de manchas varia de 0 a 528, contudo quando não há número disponível para o dia, o valor é registrado com -1.

O *dataset Parking Birmingham* (PB) (STOLFI; ALBA; YAO, 2017) (Figura 11f) são dados da taxa de ocupação do parque *Birmingham*, operados pelo *National Car Parks* (NCP) do *Birmingham City Council*. *Parking Birmingham* possui 4 subséries, das quais foi adquirida a subsérie “*Occupancy*” que é a variável numérica principal da série, enquanto as outras três são as datas e capacidade do estacionamento e a *string* de Número do código do sistema.

Daily Minimum Temperatures in ME (DMTME) (KANKANA, 2022)(Figura 11g) são dados de registros de temperaturas mínimas diárias em *Maine* EUA, no período de 1981-1990.

Electricity Usage (EU) (DUNN, 2018) (Figura 11h) são dados da quantidade de quilowatts-hora consumido em uma residência durante um período de 3,5 meses, de 25 de novembro de 2011 a 17 de março de 2012. A série possui 4 variáveis (“*Consumption*”, “*kWh*”, “*Off-peakMid-peak*”, “*On-peak*”) com uma codificação para uso no pico (10,8c/kWh), no pico médio (9,2c/kWh) e fora do pico (6,2c/kWh). Apenas a variável “*Consumption*” foi selecionada.

O *dataset Electricity Load Diagrams 20112014* (ELD)(TRINDADE, 2021)(Figura 11i) contém dados de consumo de energia referente ao período de 2011-2015. As mostras foram coletas a cada 15 segundos em um total de 140.256 amostras com valores variando de 0 a 235,625. Esse *dataset* possui 370 variáveis (MT-001: MT-370), onde MT-001 é a série de consumo referente de número 1, totalizando 370 séries de 370 clientes.

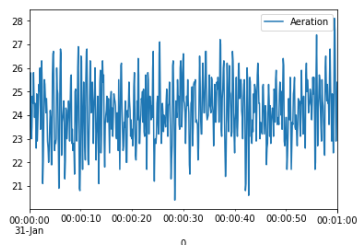
Paper basis weight (PBW) (DUNN, 2018) (Figura 11j) é uma série de dados da medida da gramatura seca de papel em uma indústria. Essas medições são de um medidor de varredura *on-line*, feito a cada 30 segundos. Os dados possuem 231 amostras.

Ruber Colour (RC) (DUNN, 2018)(Figura 11k) são dados da cor de um determinado produto de borracha. Esta série é um exemplo de dados industriais para demonstrar

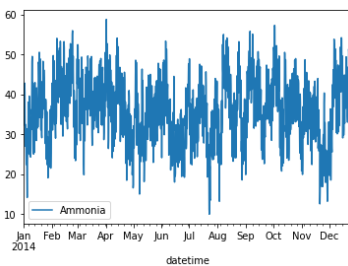
como construir um gráfico de monitoramento da cor de um produto de borracha. A série de dados possui 100 amostras com valores variando entre 220 e 255.

Figura 11 – Gráfico das séries estacionarias

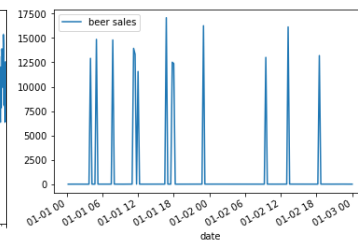
(a) Gráfico da série *Aeration rate*



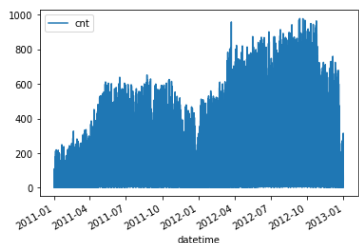
(b) Gráfico da série *Ammonia*



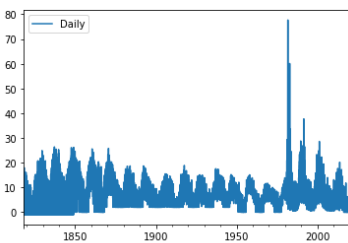
(c) Gráfico da série *Beer sales*



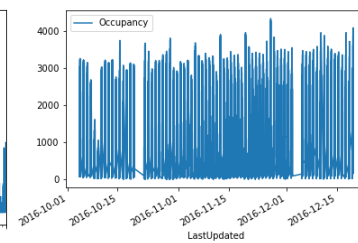
(d) Gráfico da série *Bike sharing*



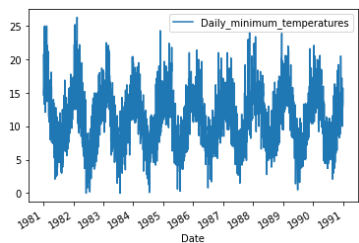
(e) Gráfico da série *Sunspots*



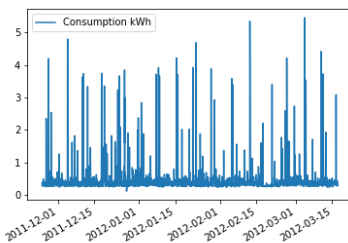
(f) Gráfico da série *Parking Birmingham*



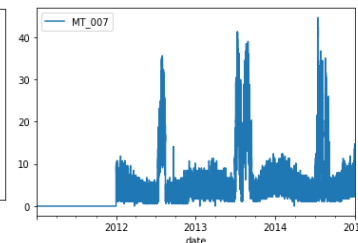
(g) Gráfico da série *Daily minimum temperatures in ME*



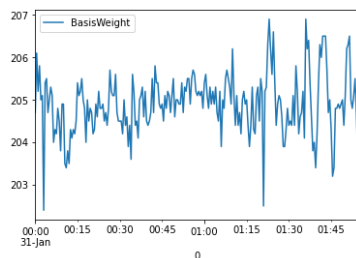
(h) Gráfico da série *Electricity usage*



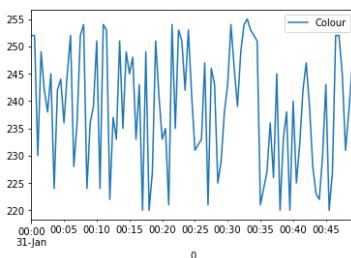
(i) Gráfico de uma série do dataset *ElectricityLoadDiagrams 20112014*



(j) Gráfico da série *Paper basis weight*



(k) Gráfico de uma série do *Rubber colour*



Fonte: Acervo do autor.

Tabela 3 – Principais características dos *datasets* das séries estacionárias

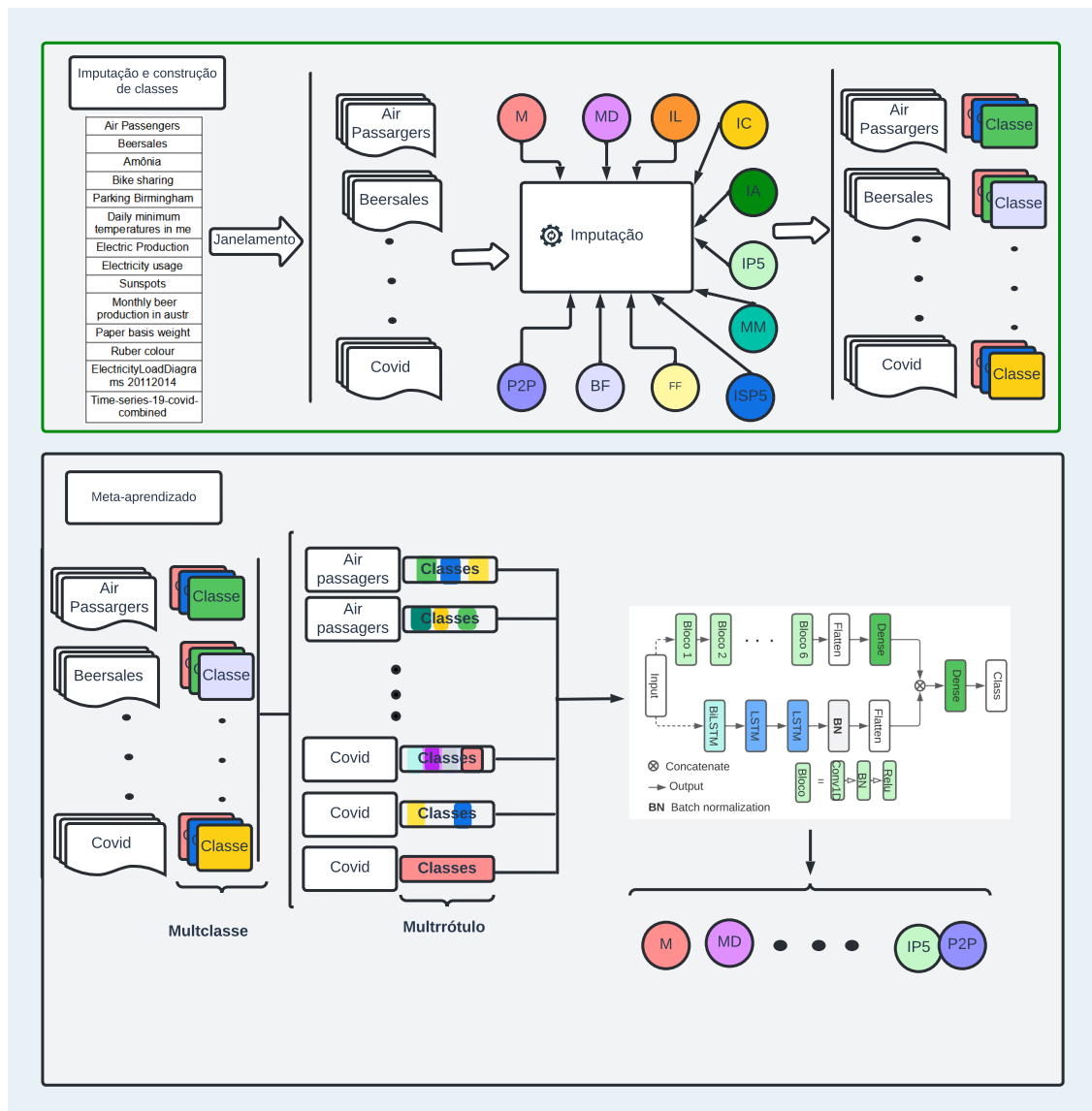
Dataset	Fonte	Quantidade de séries	Intervalo de valores
<i>Aeration rate</i>	(DUNN, 2018)	1	[20,4 , 28,1]
<i>Ammonia</i>	(DUNN, 2018)	1	[9,9 , 58,4]
<i>Beer sales</i>	(GRAVES, 2022)	1	[9,8 , 17081]
<i>Bike sharing</i>	(FANAEE-T; GAMA, 2013)	1	[1, 977]
<i>Sunspots</i>	(DATA/IMAGE; BELGIUM; BRUSSELS, 2021)	1	[0 , 26]
<i>Parking Birmingham</i>	(STOLFI; ALBA; YAO, 2017)	1	[1, 4327]
<i>Daily Minimum Temperatures in ME</i>	(KANKANA, 2022)	1	[0 , 26,3]
<i>Electricity Usage</i>	(DUNN, 2018)	1	[0,12 , 5,45]
<i>Electricity Load Diagrams 20112014</i>	(TRINDADE, 2021)	370	[0, 4578125]
<i>Paper basis weight</i>	(DUNN, 2018)	1	[202,4 , 206,9]
<i>Ruber Colour</i>	(DUNN, 2018)	1	[220, 255]

4.2 Método Proposto

O método desenvolvido neste trabalho visa auxiliar no processo de imputação de dados faltosos em séries temporais univariadas. Para isso, foram, inicialmente, selecionados *datasets* públicos e univariados. Esses *datasets* foram filtrados, onde apenas as séries que possuíam variáveis numéricas foram selecionadas. A principal contribuição do método proposto está na proposta de uma rede para usar no meta-aprendizado (VANSCHOREN, 2018) para imputar dados ausentes em séries temporais.

As etapas do método proposto baseado em meta-aprendizado consistem em: aquisição e pré-processamento da base de dados da STU, imputação dos dados faltosos simulados usando diferentes métodos de imputação, avaliação dos resultados das imputações e construção do metadado, desenvolvimento da rede, treinamento da rede proposta. Nas próximas seções o método será detalhada. As principais etapas estão representadas na Figura 12 e serão descritas nas próximas seções.

Figura 12 – Etapas do método proposto.



Fonte: Acervo do autor.

4.2.1 Pré-processamento

O método proposto (Figura 12) inicia-se com a aquisição dos *datasets*, descritos na seção 4.1, para construção de um banco de dados construído a partir da mineração de séries temporais univariadas.

Durante a seleção dos *datasets* foi observado que embora tivessem a descrição do período de observação temporal de algumas séries, os dados originais não possuíam um campo com essa informação, ou seja, os *Comma-separated values* (CSVs) possuíam apenas as colunas com as observações sem a marcação temporal, muito embora houvesse a descrição dessa marcação temporal no local de hospedagem das séries. Por isso, para essas

séries, foram produzidas horas e datas referentes a cada observação utilizando a biblioteca *datetime 3.10* (ROSSUM; AL., 2021). Além disso, algumas anomalias foram corrigidas, anomalias como tipo de dados numéricos sendo lidos como *strings*, orientações diferentes entre séries, séries com cabeçalho e outras sem, e anomalias geradas pelo próprio sistema de aquisição de dados, produzindo amostras a mais.

Em seguida, realiza-se o janelamento, no qual a partir de uma série se produziu um conjunto de subséries com mesmo tamanho. Além disso, com objetivo de aproveitar a maior quantidade de dados possível, mas sem repetir todos os elementos, as janelas que não possuíam elementos suficientes para se igualar ao tamanho de janela definido foi descartada quando não possuiu pelo menos metade das janelas, e as que possuíram pelo menos metade, foi redimensionada, considerando a inversão da série. Assim, nesse caso, pelo menos duas janelas possuiu 50% dos elementos comuns.

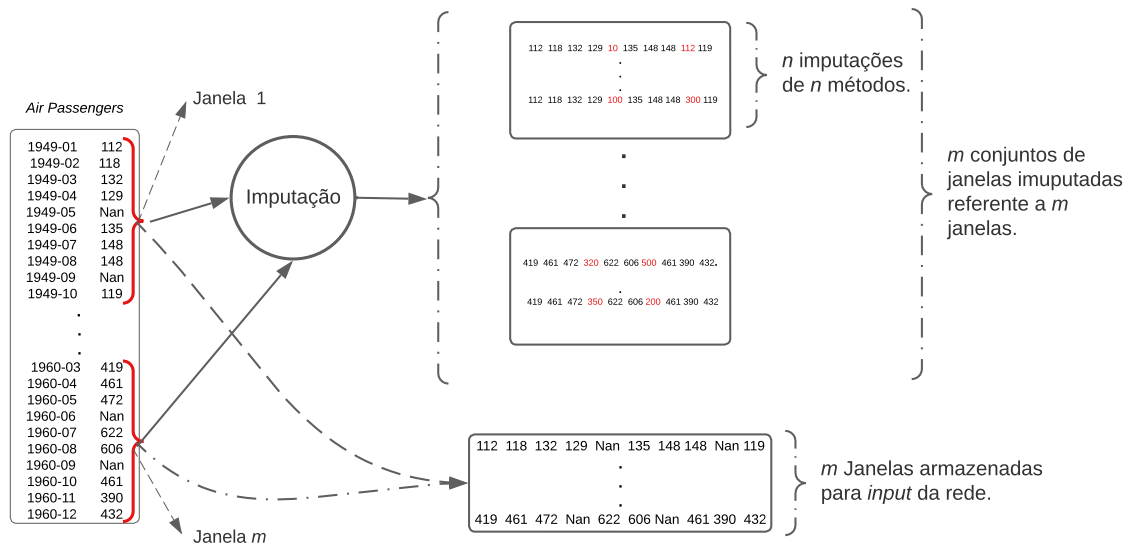
Por fim, foram removidos aleatoriamente valores das janelas com taxas de 10%, 20% e 30%. As taxas são referentes ao tamanho das janelas e não das séries. Esse processo pode ser visto no início da Figura 13 na Seção 4.2.2.

4.2.2 Imputação

No passo da imputação, tem-se um exemplo com a série *Air passengers* mostrado na Figura 13. Inicialmente a série é dividida em m janelas de determinado tamanho, então são removidos aleatoriamente amostras das janelas, então a janela com valores faltosos é apresentada a primeira técnica de imputação que imputa e retorna a janela com os valores imputados. A mesma janela é apresentada à uma segunda técnica de imputação com os mesmos valores faltosos e , assim por diante até que a mesma janela seja apresentada a todas as técnicas.

Para cada técnica de imputação temos uma janela com valor imputado que são todas empilhadas. Na Figura 13 vê-se estas janelas com os valores imputados em vermelho, pode-se verificar que apenas os valores em vermelho mudam, indicando uma imputação diferente por uma técnica. Paralelamente à imputação as janelas com os valores perdidos, também são armazenadas empilhadas para os passo seguinte do método, verifica-se na Figura 13 com a caixa com a descrição “ m Janelas armazenadas para *input* da rede”. Os métodos de imputação utilizados foram explicados na Subseção 3.3. Além desses, propôs-se um uma técnica de imputação usando Pix2Pix GAN descrita na próxima Subseção 4.2.2.1.

Figura 13 – Exemplo detalhado do método até a imputação com a série *Air Passengers*

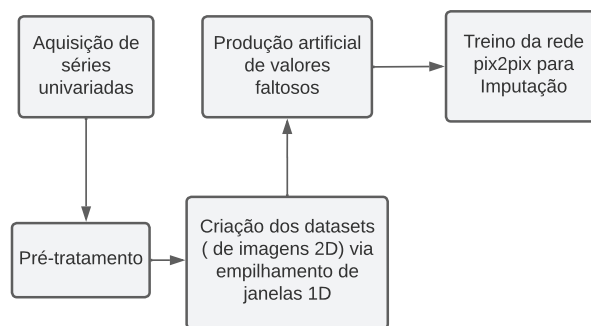


Fonte: Acervo do autor

4.2.2.1 Técnica de Imputação Utilizando Pix2Pix

A partir da rede Pix2Pix descrita na Subseção 3.3, propôs-se um novo método para imputar dados ausentes em ST, ilustrado na Figura 14, que se inicia pela aquisição de *datasets* univariados e públicos, os quais são adequados sob um mesmo padrão (disposição em linhas), para depois remover valores aleatoriamente e, em seguida, aplicar a rede Pix2Pix para recuperar esses valores artificialmente perdidos (ALMEIDA et al., 2023).

Figura 14 – Etapas do método proposto usando a rede Pix2Pix.

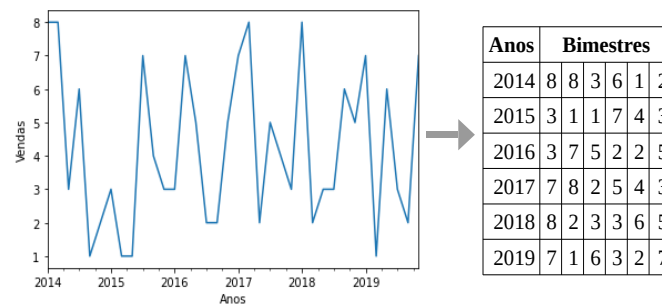


Fonte: Adaptado de (ALMEIDA et al., 2023).

Realizado o pré-processamento, inicia-se a construção das imagens 2D. Para converter Série Temporal Univariada (STU) 1D em imagem 2D, propôs-se um novo método baseado no trabalho de Zhuang, Ke e Wang (2019). Para efeito de comparação, o estudo separou dois bancos de imagens, um com uma transformação simples (T1) que apenas reorganiza sequencialmente, em um formato 32x32, janelamentos de tamanho 1.024, e o método de transformação proposto (T2), que consiste em dispor a série sob um mesmo padrão temporal. Dessa forma, realizou-se a discriminação das subséries com a mesma marcação de tempo e data, de forma que essas subséries foram truncadas ao se obter 32 elementos. Quando se obtém 32 subséries com a mesma marcação temporal com diferenças apenas do ano, obtém-se uma imagem 32x32.

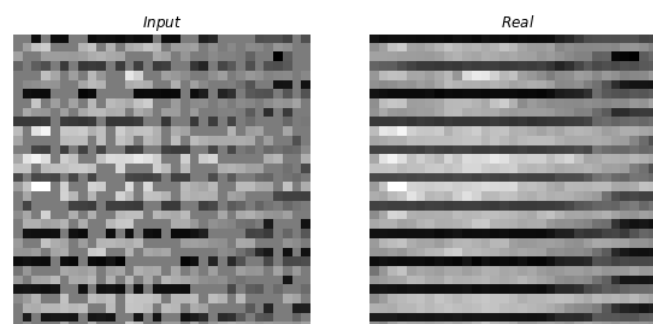
Suponha-se, por exemplo, as vendas bimestrais de uma determinada corretora de imóveis entre os anos de 2014 e 2019. Ao dispor essas vendas sob a mesma marcação temporal, obtemos para cada linha e coluna, o ano de vendas e o bimestre, respectivamente. Pode-se observar esse processo na Figura 15.

Figura 15 – Transformação proposta.



Fonte: Adaptado de (ALMEIDA et al., 2023).

Após as transformações, foram inseridos valores faltosos à taxa de 40%. Como a rede não aceita valores não numéricos, foi necessário representar os valores faltosos com um valor diferente de todos os que a imagem possuía, por isso testou-se substituir, nas porcentagens mencionadas, por 0, $[\min -1]$, $[\max +1]$ e média. Assim, de todos os de preenchimento de valores perdidos citados, os valores (0, $[\min -1]$, $[\max +1]$ e média) reduzem a variação no conjunto de dados. Por outro lado, a média é a única que mantém a posição da distribuição dos dados. Min e Max são muito sensíveis a valores discrepantes (*outliers*) do conjunto de dados. Pode-se verificar um exemplo na Figura 16, onde os valores faltosos (*Input*) são representados por *pixels* diferentes dos valores reais (*Real*).

Figura 16 – Exemplo de imagem com (*Input*) e sem (*Real*) valores faltosos.

Fonte: Adaptado de (ALMEIDA et al., 2023).

Com isso, parte-se para o treinamento da rede. A rede Pix2Pix (ISOLA et al., 2016) foi desenvolvida para solução de transformação de imagem para imagem. Essa transformação parte geralmente de um esboço, sejam bordas ou mapas de características. Também obteve sucesso em colorir imagens ou desenhos de objetos em geral.

A Pix2Pix foi construída a partir dos modelos de Redes Adversárias Generativas (GANs), as quais buscam em seu treinamento estabelecer um jogo entre duas redes neurais

denominadas de geradora e discriminadora (PAN et al., 2019). Na Pix2Pix, utiliza-se a U-net (RONNEBERGER; FISCHER; BROX, 2015b) como rede geradora, por tentar gerar imagens reais a partir de imagens falsas. Uma rede *encoder* recebe o nome de rede discriminadora, no qual a partir da concatenação da imagem gerada pelo gerador e a imagem real, tenta aprender se a imagem do gerador é a imagem verdadeira.

O fluxo de aprendizado da rede consiste no gerador tentar enganar o discriminador e esse por sua vez tenta aprender a diferenciar uma imagem real de uma imagem falsa. Com isso, foi proposta modificações no gerador e decodificador. Para o gerador, propomos uma versão modificada da U-net (Figura 17), pois para aproveitar melhor as séries disponíveis, a entrada foi modificada para 32x32x1 em vez de 256x256x3 da versão original. Além disso, o decodificador foi modificado para as dimensões dos *kernels* de 4 para 2, Figura 18.

Figura 17 – Gerador proposto.

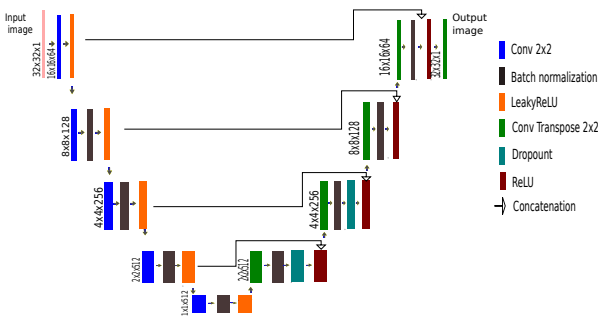
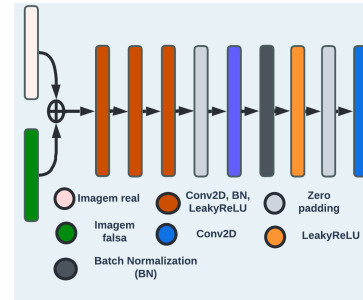


Figura 18 – Decodificador proposto.



Fonte: Adaptado de (ALMEIDA et al., 2023).

Adicionalmente, foi feita a mudança da função de ativação do último bloco de convolução do gerador, trocando a Tangente hiperbólica para Sigmoide. Além disso, as entradas da rede foram normalizadas via normalização *Max-Min* (Eq. 4.1), pois assim, a maioria dos blocos da rede são ativados via *Sigmoide* e a função de custo da rede é uma composição com a mesma, e por isso, se entendeu que a normalização *Max-Min*, que normaliza entre 0 e 1, fosse a mais adequada. Na última fase, teste da rede, as métricas de avaliação são aferidas apenas para os valores em que todas as técnicas conseguiram imputar.

$$I_{i*} = \frac{I - \min\{I\}}{\max\{I\} - \min\{I\}}, \quad (4.1)$$

onde I_{i*} é a imagem normalizada, I a imagem real e max e min a função máximo e mínimo, respectivamente.

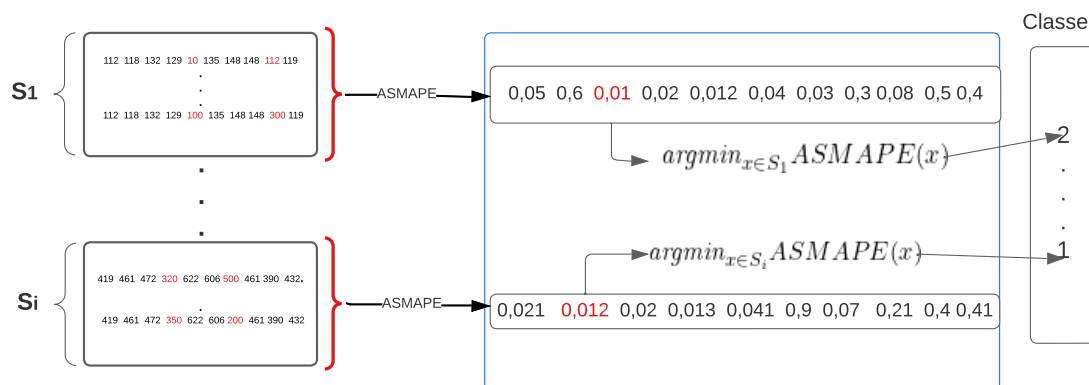
Por fim, nesse estudo não será abordada a transformação proposta, visto que mesmo com a transformação simples (T1) a rede obteve resultados que também superaram em alguns *datasets* os métodos comparados. A transformação poderá beneficiar o método de imputação proposto e prejudicar os demais, visto que transforma a série em sua essência, pois além de descartar muitas amostras a organização temporal é mudada.

4.2.3 Construção do Metadado

A fase seguinte a imputação (Subseção 4.2.2) é a construção do metadado, ilustrada na Figura 13. Nessa etapa, para construir o metadado são consideradas as janelas armazenadas durante o processo de imputação e as janelas com valores imputados.

As janelas armazenadas não sofrem mais alterações, são apenas guardadas em arquivo *Comma-separated values* (CSV). A partir das janelas imputadas, é calculado o ASMAPE de cada método e em seguida armazena-se a classe referente ao método que obteve o menor ASMAPE. As técnicas sempre imputam na mesma ordem, onde atribui-se um rótulo a cada algoritmo de imputação e esse rótulo é utilizado na construção do metadado, rotulando as séries com a melhor técnica de imputação definida durante a etapa (Seção 4.2.3).

Figura 19 – Exemplo detalhado da construção do metadado até a geração dos rótulos



Fonte: Acervo do autor

Contudo, notou-se durante os experimentos que o uso de multi-classe, não se adequou bem a nosso problema. De acordo com Ma e Feng (2017), Zadeh, Sayadi e Kosari (2019) a abordagem multi-rótulo pode ser mais adequada para lidar com séries temporais devido à sua capacidade de lidar com múltiplas classes simultaneamente e capturar a correlação entre elas. No entanto, a escolha do método de classificação depende do conjunto

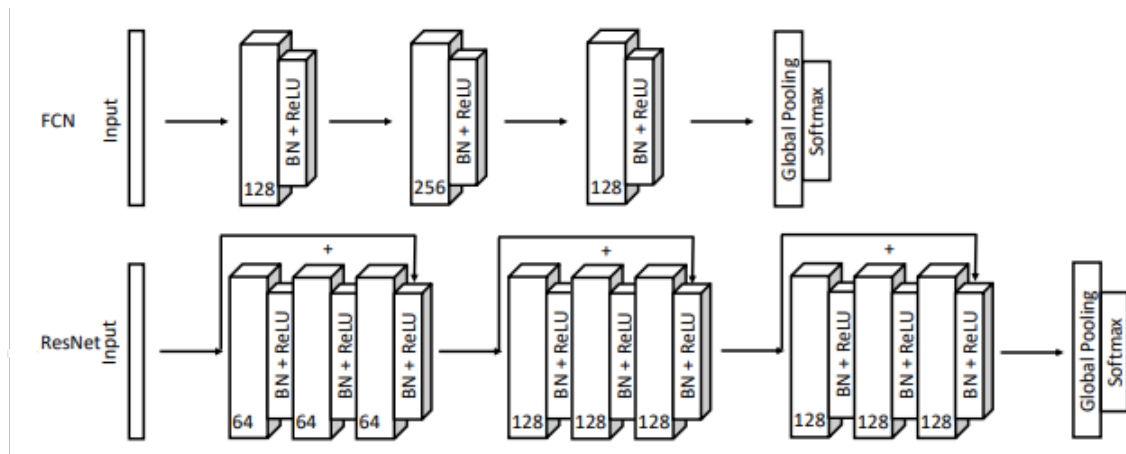
de dados e das características específicas do problema em questão, portanto, foi importante avaliar cuidadosamente a abordagem mais adequada para nossa tarefa. E realizando os mesmos experimentos alterando com multiclasse e multirótulo os resultados se mostraram melhores para o caso multirótulo. Assim, pode-se inferir que devido a quantidade de classe objetivo serem maiores há ganho na probabilidade da rede acertar, ou seja, acertar escolhendo um conjunto de técnicas é mais provável do que em um único elemento. Portanto, adotado para o treinamento da rede a abordagem multi-rótulo.

Por isso, após a imputação, as técnicas que em determinada janela tiver ASMAPE abaixo de 2% são marcadas como classe objetivo, ou seja, como rótulo. Caso o menor ASMAPE da janelas seja maior que 2,5 % se escolhe o menor ASMAPE. Em seguida, as mesmas janelas são apresentadas à entrada da rede que treina baseada nas classes (métodos) com melhor desempenho. Essa abordagem permite que vários métodos possam ser usados para imputar valores na mesma série em trechos diferentes, pois a imputação de dados nos mostra que um mesmo método de imputação nem sempre é o mais apropriado para preencher diferentes lacunas em uma mesma série temporal, pois a imputação de dados nos mostra que um mesmo método de imputação nem sempre é o mais apropriado para preencher diferentes lacuna em uma mesma série temporal.

4.2.4 HybridLSTM: Rede proposta para classificação

A rede proposta para classificação é uma rede híbrida baseada inicialmente em Wang, Yan e Oates (2016). Dessa rede utilizou-se a ideia do bloco presente na ResNet e *fully convolutional networks* (FCN) que é composto de normalização de *batch* e ativação *Relu*.

Figura 20 – Rede Base para a HybridLSTM

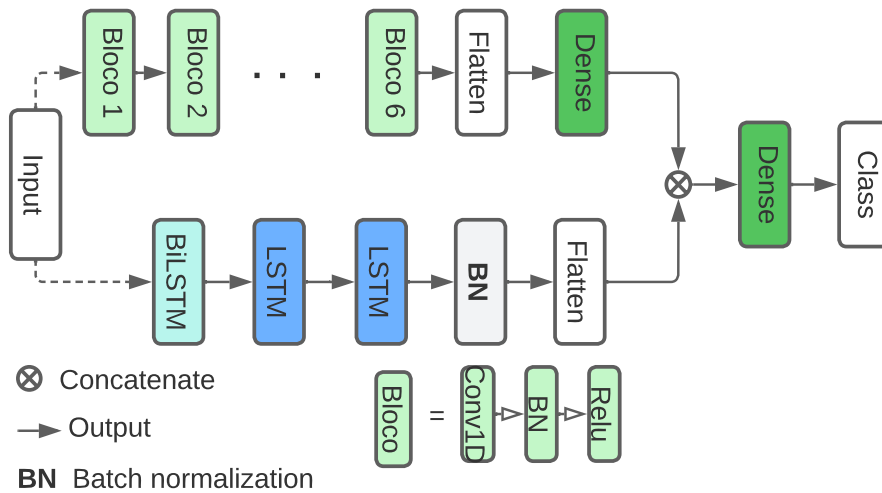


Fonte: Adaptado de (WANG; YAN; OATES, 2016)

Assim na rede proposta adicionou-se uma convolução ao bloco. Ela possui dois grandes blocos principais, onde a entrada toma dois caminhos diferentes para ser concatenada no final (vide Figura 21). O primeiro caminho possui 6 blocos compostos por uma convolução 1D, uma Normalização de Baeth e uma função de ativação *ReLU*. Os blocos de 1 a 6 são responsáveis, principalmente, pela extração de características espaciais e visuais, as quais são reordenadas e filtradas por um bloco denso. Esse caminho é composto por uma rede que busca extrair principalmente características atemporais.

O segundo caminho da rede é composto por um bloco de rede BiLSTM, a qual é uma rede com memória maior que a LSTM, por ter dois blocos LSTMs que trabalham em conjunto compartilhando filtros e aprendizado. Logo após passar pelo bloco BiLSTM os dados passam por dois blocos LSTM para confirmar e adicionar os dados de memória média, longa e curta. Assim, os resultados são normalizados e combinados com a saída da primeira rede e passam para um bloco de classificação. Essa segunda rede consegue extrair e relacionar características das séries por um período maior, possibilitando um melhor aprendizado no que diz respeito a características temporais, uma vez que as LSTMs são redes eficientes nesse tipo de análise.

Figura 21 – Rede Proposta



Fonte: Acervo do autor

Essa rede usa a função de perda *BinaryCrossentropy*, pois ela trata cada classe separadamente das outras, sendo mais adequada ao problema, *multilabel* e, por conta dessa função, necessita-se de uma métrica adequada para a classificação binária das classes e por isso a métrica mais adequada é a *Area under the ROC Curve* (AUC) (ALHARTHI et al., 2021), que avalia as variáveis *true_positives*, *true_negatives*, *false_positives* *false_negatives*, obtendo ainda a vantagem de aproveitamento dos blocos de ativação por *sigmoide* e otimizando o trabalho do gradiente por aceitar *thresholds* diferentes para cada classe.

5 RESULTADOS E DISCUSSÃO

Este capítulo apresenta e discute os dados pertinentes aos experimentos realizados, bem como resultados obtidos em cada etapa do método proposto para imputação de valores faltosos em séries temporais. Avaliou-se o resultado das imputações por técnica de imputação, as imputações realizadas por meio da média e mediana, assim como o resultado final do método baseado em meta-aprendizado, alcançado por meio da rede HybridLSTM. Por fim, discute-se o resultado final obtido pelo método proposto e como esse trabalho contribui para a resolução do problema de dados faltosos em séries temporais.

5.1 Hardware e Softwares Utilizados

O método apresentado no Capítulo 3 foi desenvolvido por completo na linguagem de programação *Python*, na versão 3.9.12. Esta linguagem foi escolhida por oferecer grande suporte ao desenvolvimento de sistemas baseados em Aprendizado de Máquina, e ao processamento e análise de dados científicos. Além disso, o python tem sido a linguagem mais utilizada pela comunidade científica e por sistemas *Linux* e por isso conta com grande suporte. Dentre as principais bibliotecas em *Python* usadas no desenvolvimento desta pesquisa estão: *Keras* (CHOLLET et al., 2015), na versão 2.9.0 e *Tensorflow* na versão 2.9.2, *Pandas* na versão 1.4.3 (MCKINNEY et al., 2011) e *scikit-learn* (KRAMER, 2016) na versão 1.0.2. O hardware utilizado para o desenvolvimento da pesquisa conta com sistema operacional *Windows 10*, com processador Intel Core i3-7700, 4 GB de memória RAM. Para o treinamento da rede proposta, utilizou-se a GPU NVIDIA 1080Ti com *frame buffer* de 11 *gigabytes*

5.2 Desempenho das Técnicas de Imputação Individuais

Inicialmente, um experimento foi realizado para avaliar a imputação individual das técnicas avaliadas nesse estudo. Para tanto, foram selecionadas apenas as séries com pelo menos 1.024 observações (Seção 4.2). Essa delimitação foi necessária devido à entrada da rede Pix2Pix ser 32 x 32, resultando em 5 *datasets* com 374 séries, subdividas em 50.713 janelas. Além dessa filtragem, não foram utilizadas janelas de séries que possuíam apenas zeros, pois algumas séries de dados de consumo de energia do *dataset* ELD (Seção

4.1.2), nos meses iniciais, não possuíam dados de consumo e isso foi registrado pela própria empresa como consumo zero. Após organização das janelas, realizou-se a imputação dos dados faltosos utilizando todas as técnicas avaliadas nesse estudo.

No que diz respeito às técnicas de imputação, para a reduzir o custo computacional, neste estudo realizou-se um experimento inicial para se obter as melhores variações das técnicas utilizadas. As interpolações de grau 2 a 7 foram avaliadas, apenas a interpolação polinomial de grau 3 e 5 foram selecionadas. Para as *splines* foram avaliadas as ordens 1 e 5 e selecionada a de ordem 5. Por fim, avaliou-se o tamanho das janelas deslizantes da média móvel com tamanho entre 1 e 10, selecionou-se o com 3 elementos.

A Figura 22 apresenta a frequência percentual de cada técnica em relação às taxas de valores faltosos, a saber 10% 20% e 30%. Essa frequência é o percentual de janelas na qual as técnicas obtiveram o menor ASMAPE na imputação. Assim, pode-se notar que a interpolação linear (IL), Akima (IA) e a rede Pix2Pix (P2P) superaram o restante das técnicas com a maior estabilidade para a rede P2P, que teve a frequência muito semelhante nos três experimentos. Por outro lado, a média (M) e a interpolação quártupla (IP5) obtiveram as menores frequências. Contudo, esses resultados confirmam não haver supremacia de uma técnica sobre outra em todos os tipos de séries, por isso levanta-se a hipótese desse padrão também está presente em toda a extensão de uma mesma série.

Analisando os resultados apresentados na Tabela 4, confirma-se a hipótese de que uma técnica não é melhor em toda a extensão de uma série, pois se observa que em 5 datasets distintos, ao mudarmos a taxa de valores faltosos e conseqüentemente a posição dos valores faltosos em uma mesma série, há variação no desempenho da imputação de um mesma técnica em uma mesma série. Isso ocorre, por exemplo, nas séries do dataset ELD, denominadas MTs, referentes a três padrões de energia distintos. Na Tabela 4, observa-se que na técnica IL a frequência absoluta das janelas aumenta, onde ela domina na imputação, conforme aumenta a taxa de valor faltoso, enquanto a técnica BF diminui, ou seja, na série MT_001 o método IL supera todos os métodos a 10% em 13 janelas e vai para 20 janelas a 20% enquanto BF cai de 19 janelas para 12.

Figura 22 – Frequência percentual da melhor técnica de imputação definida com base na métrica ASMAPE em 374 séries.

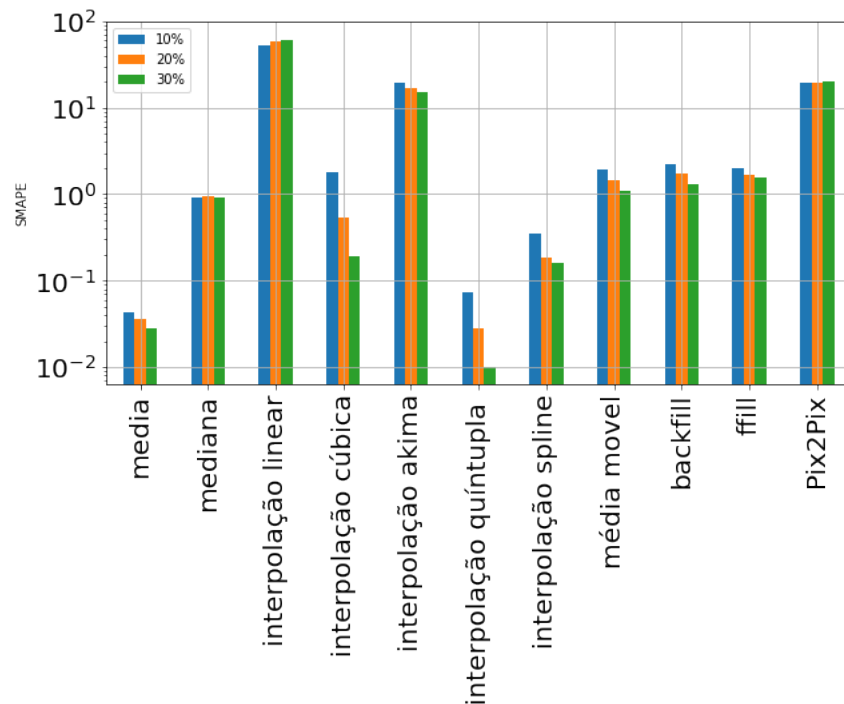


Tabela 4 – Frequência absoluta por técnica de imputação em 5 *datasets* com base no ASMAPE

Taxa	Dataset	M	MD	IL	IC	IA	IP5	IsP5	MM	BF	FF	P2P
10%	ELD	22	460	26458	896	9699	35	180	991	1125	1008	9812
	AMM	0	0	0	0	0	2	0	0	0	0	0
	BIK	0	0	5	2	10	0	0	0	0	0	0
	DAY	0	0	4	0	0	0	0	0	0	0	0
	SUP	0	0	4	0	0	0	0	0	0	0	0
Total		22	460	26471	898	9709	37	180	991	1125	1008	9812
20%	ELD	18	488	29107	269	8441	12	93	726	887	841	9807
	AMM	0	0	0	0	0	2	0	0	0	0	0
	BIK	0	0	4	0	13	0	0	0	0	0	0
	DAY	0	0	4	0	0	0	0	0	0	0	0
	SUP	0	0	4	0	0	0	0	0	0	0	0
Total		18	488	29119	269	8454	14	93	726	887	841	9807
30%	ELD	14	464	30233	97	7703	3	81	561	666	776	10091
	AMM	0	0	0	0	0	2	0	0	0	0	0
	BIK	0	0	2	0	15	0	0	0	0	0	0
	DAY	0	0	4	0	0	0	0	0	0	0	0
	SUP	0	0	4	0	0	0	0	0	0	0	0
Total		14	464	30243	97	7718	5	81	561	666	776	10091

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), spline (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF) e Pix2Pix (2P2).

Assim, evidencia-se o principal objetivo da rede hybridLSTM (Seção 21) que é reconhecer as características das janelas para recomendar a melhor técnica ou modelo de

imputação. Dessa forma, a rede proposta otimiza a imputação em uma série escolhendo diferentes técnicas para as janelas que compõem a STU.

A Tabela 5 mostra a descrição dos resultados referentes aos experimentos com 10%, 20% e 30% de dados faltosos. Nota-se que para estes experimentos, no terceiro quartil, 75% dos dados, não ha variação significativa na métrica MAE, ou seja, todas as técnicas obtiveram desempenho semelhante em todos os experimentos com variação menor 1 desvio padrão, quando comparamos 50% das amostras a 75%.

Tabela 5 – Descrição dos resultados referente a métrica MAE.

		M	MD	IL	IC	IA	IP5	ISP5	MM	BF	FF	P2P
Experimento 10 %	Média	82,66	78,83	10,89	12,48	11,02	14,78	15,05	16,22	14,91	14,82	2,86
	std	188,70	180,26	19,11	21,73	19,29	25,68	26,58	29,37	26,52	26,36	1,84
	Mínimo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,29
	25%	5,11	4,86	1,42	1,63	1,45	1,91	1,63	1,96	1,86	1,86	1,41
	50%	27,58	26,13	4,64	5,37	4,71	6,37	6,17	6,64	6,22	6,18	2,93
	75%	72,11	68,59	10,98	12,81	11,18	15,21	15,29	15,90	14,75	14,71	3,67
	Máximo	3.061,89	3.293,00	206,25	183,16	187,73	200,55	199,24	458,98	340,68	346,49	27,72
Experimento 20 %	Média	79,57	76,02	10,86	12,59	11,02	15,17	15,61	15,83	15,12	14,73	3,02
	std	180,15	172,43	18,66	21,51	18,88	25,87	27,11	28,02	26,36	25,72	1,70
	Mínimo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,28
	25%	4,97	4,72	1,46	1,70	1,50	2,02	1,70	1,97	1,94	1,90	1,61
	50%	27,28	25,88	4,74	5,57	4,84	6,75	6,53	6,67	6,48	6,32	3,29
	75%	70,76	67,63	11,13	13,11	11,39	15,84	16,05	15,85	15,25	14,81	3,94
	Máximo	2.899,84	2.829,55	255,59	223,62	213,86	200,82	199,58	508,25	294,56	508,01	23,94
Experimento 30 %	Média	75,30	71,99	10,67	12,51	10,84	15,40	16,32	15,20	15,25	14,34	9,52
	std	166,83	159,88	17,83	20,82	18,06	25,55	27,86	26,07	26,08	24,28	3,27
	Mínimo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,83
	25%	4,76	4,55	1,46	1,74	1,51	2,12	1,71	1,96	1,97	1,90	7,11
	50%	26,89	25,54	4,82	5,72	4,92	7,05	6,90	6,67	6,73	6,39	9,63
	75%	68,78	65,66	11,18	13,35	11,45	16,51	17,00	15,64	15,71	14,91	12,14
	Máximo	2.805,86	2.559,75	181,75	237,59	170,64	254,86	200,00	262,87	361,59	233,85	28,65

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), spline (IS), akima (IA), polinomial quántupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF) e Pix2Pix (2P2).

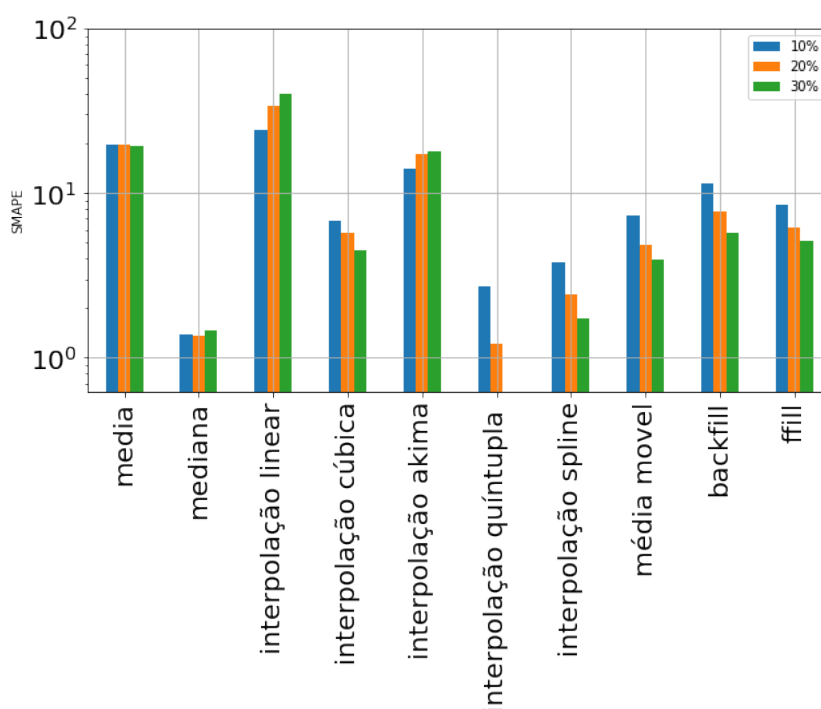
Contudo, a ISP5 obtém um desvio padrão muito alto, mas observa-se que 50% das amostras são MAE com valor menor que 12 e isso implica dizer que esse método obteve baixo desempenho em janelas específicas de algumas séries. Observado o valor máximo imputado pela média, pode-se relembrar também que a maioria dessas 374 séries possuem valores na classe de milhares.

Por outro lado, foi necessário realizar um experimento com uma variação maior de séries, pois os estudos levantados (Seção 2) são limitados na quantidade de séries de distintas áreas e, por isso, acredita-se que é uma lacuna a ser preenchida, logo realizou-se a seleção de todas as séries que possuem ao menos 100 observações, resultando em 776

séries e 524.397 janelas. Assim, não foi possível nesse experimento incluir a rede Pix2Pix, ficando a comparação entre as outras dez técnicas de imputação.

Na Figura 23 observa-se que com as 776 séries o balanceamento foi mais regular do que com apenas 374 séries (Figura 22). A exceção foi a interpolação quártupla que, com 30% de dados faltosos, conseguiu superar todos as outras técnicas comparadas em apenas 0,624% das janelas. Nota-se que a interpolação linear, Akima e cúbica obtiveram resultados melhores com mais dados faltosos, enquanto os demais se mantiveram regulares ou obtiveram resultados melhores com menos dados faltosos.

Figura 23 – Frequência percentual da melhor técnica de imputação definida com base na métrica ASMAPE em 776 séries.



A Tabela 6 mostra a frequência absoluta onde cada técnica obteve melhor imputação em função da métrica ASMAPE nas janelas geradas. Os dados da tabela confirmam que em trechos diferentes de uma mesma série há variação de técnicas com melhor imputação. Nota-se, ainda, que a interpolação linear (IL) foi a melhor técnica nos três experimentos, seguida respectivamente pela média (M) e interpolação Akima (IA). O desempenho de uma técnica simples como a média pode estar relacionado ao tamanho do janelamento e tipo de série, pois nos datasets de casos de COVID e consumo de energia elétrica (ELD) esperava-se tal desempenho, uma vez que para o tamanho da janela escolhido a média é um valor razoável para imputar.

Tabela 6 – Descrição do desempenho por técnica e dataset referente a métrica ASMAPE

Datasets		M	MD	IL	IC	IA	IP5	ISP5	MM	BF	FF
Experimento 10 %	AirPassengers	0	0	0	1	0	1	0	0	0	0
	Birmingham Parking	0	0	13	78	100	13	22	0	0	1
	Electric_Production	0	0	0	0	0	3	1	0	0	0
	ELD	102605	7291	126183	34791	72712	13962	19742	38021	59792	44011
	Sp	0	0	1	0	1	0	0	0	0	0
	aeration_rate	0	0	1	3	0	1	0	0	1	0
	ammonia	0	0	0	3	0	10	2	0	0	0
	bike_sharing	0	0	51	40	57	9	10	1	3	3
	covid-combined	804	39	1327	410	843	159	205	125	471	391
	daily_minimum_temperatures_in_me	2	0	9	5	7	1	7	3	1	2
	manchasol	0	1	13	3	8	0	2	1	5	4
	monthly_beer_production_in_austr	1	1	1	0	0	0	0	0	0	2
	paper-basis-weight	1	0	1	0	0	0	0	0	1	0
	rubber-colour	0	0	1	0	0	0	0	0	0	0
Total	103413	7332	127601	35334	73728	14159	19991	38151	60274	44114	
Experimento 20 %	AirPassengers	0	0	1	1	0	0	0	0	0	0
	Birmingham Parking	0	0	3	101	110	11	13	0	0	1
	Electric_Production	0	0	0	1	0	0	3	0	0	0
	ELD	102608	7140	174252	29425	89202	6290	12685	25076	40362	32070
	Sp	0	0	1	0	1	0	0	0	0	0
	aeration_rate	1	1	2	2	0	0	0	0	0	0
	ammonia	0	0	0	5	2	8	0	0	0	0
	bike_sharing	0	0	44	31	91	4	4	0	0	0
	covid-combined	792	26	1391	444	989	119	122	130	402	359
	daily_minimum_temperatures_in_me	3	1	12	4	8	0	5	1	2	1
	manchasol	4	2	17	3	2	2	1	0	4	2
	monthly beer production in austr	0	1	4	0	0	0	0	0	0	0
	paper basis weight	0	0	2	0	0	0	1	0	0	0
	rubber	0	0	0	0	0	0	0	0	1	0
Total	103408	7171	175729	30017	90405	6434	12834	25207	40771	32433	
Experimento 30 %	AirPassengers	0	0	0	1	0	0	1	0	0	0
	Birmingham Parking	0	0	2	138	101	6	14	0	0	0
	Electric Production	0	0	1	1	0	1	1	0	0	0
	ELD	94498	7191	194779	21415	86775	2963	8391	19441	27879	24912
	aeration rate	0	0	1	1	4	0	0	0	0	0
	ammonia	1	1	2	2	1	6	0	0	1	1
	bike sharing	0	0	53	17	96	5	1	0	0	2
	covid-19	790	20	1428	441	1073	100	78	129	329	386
	daily minimum temperatures in ME	2	4	14	2	7	2	3	1	1	1
	electricity usage	0	0	0	0	1	0	0	0	0	1
	manchasol	2	1	16	5	7	1	2	2	0	1
	monthly beer production in Australia	0	1	2	0	1	0	0	0	1	0
	Total	95293	7218	196298	22023	88066	3084	8491	19573	28211	25304

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quántupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF) e Ffill (FF) .

Tabela 7 – Descrição da distribuição descritiva da métrica MAE

		M	MD	IL	IC	IA	IP5	ISP5	MM	BF	FF
Experimento 10 %	Média	32,82	31,74	10,73	11,99	10,83	14,05	15,10	14,22	13,41	13,35
	std	38,47	38,01	19,84	21,11	19,91	23,25	25,44	23,86	22,95	22,82
	Mínimo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	25%	1,24	1,22	1,06	1,26	1,09	1,49	1,16	1,40	1,31	1,31
	50%	20,35	18,76	4,26	5,00	4,35	6,09	5,89	6,00	5,59	5,57
	75%	47,58	45,50	10,66	12,41	10,82	15,19	16,07	14,97	14,00	13,97
	Máximo	252,16	171,16	121,11	325,16	211,16	1.710,13	1.700,16	271,15	251,11	301,13
Experimento 20 %	Média	38,03	36,16	6,92	8,26	7,09	10,89	15,84	9,63	9,35	9,15
	std	47,36	45,66	10,34	12,68	10,71	18,24	63,13	13,60	13,58	13,32
	Mínimo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	25%	1,97	1,81	0,75	0,93	0,78	1,17	0,89	0,95	0,91	0,90
	50%	21,90	20,35	3,92	4,66	4,01	5,91	5,69	5,54	5,30	5,20
	75%	52,22	49,22	8,76	10,39	8,95	13,34	14,12	12,30	11,90	11,64
	Máximo	224,11	303,85	258,53	409,55	273,88	1.092,98	1.998,26	274,60	297,96	300,63
Experimento 30 %	Média	36,16	34,37	6,87	8,32	7,06	11,50	17,84	9,33	9,46	8,99
	std	44,34	42,52	10,00	12,50	10,39	20,12	71,26	12,84	13,27	12,68
	Mínimo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	25%	2,40	2,19	0,88	1,10	0,91	1,42	1,03	1,15	1,15	1,10
	50%	21,53	20,19	4,07	4,89	4,17	6,37	6,16	5,62	5,64	5,37
	75%	49,68	46,93	8,81	10,58	9,02	14,03	15,15	12,07	12,20	11,60
	Máximo	210,30	265,10	264,74	425,51	284,12	2.345,49	1.994,44	282,34	283,49	344,63

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF) e Ffill (FF) .

5.3 Resultados com Meta-Aprendizado

Para imputação de séries temporais o método de imputação proposto, incluindo a Pix2Pix, foi utilizado com os pesos originais da nossa pesquisa publicado em (ALMEIDA et al., 2023) e por esse motivo, dos *datasets* selecionados, apenas cinco foram utilizados no primeiro experimento (Seção 5.3.1), pois a entrada da rede Pix2Pix exigiu uma entrada com tamanho mínimo de 1.024 elementos. Já o segundo experimento foi realizado sem inclusão da Pix2Pix (Seção 5.3.2) para ser possível incluir uma maior variedade de *datasets*, a saber 14 *datasets*.

Os dados de entrada da rede foram balanceados usando como referência a classe da técnica Spline com a terceira menor quantidade de amostras. De acordo com Elrahman e Abraham (2013), conjuntos de dados desbalanceado afetam o desempenho das técnicas de aprendizado de máquina, pois a precisão geral e a tomada de decisão podem ser tendenciosas para a classe com mais elementos, ou classificar incorretamente as amostras da classe minoritária, ou tratando-as como ruído.

Nesses experimentos, utilizou-se a abordagem Holdout (DWORK et al., 2015),

realizando cinco execuções com divisão aleatória estratificada dos dados. A divisão dos subconjuntos de dados foi realizada na proporção de 70%, 15% e 15% para treino, validação e teste, respectivamente. Os resultados são então combinados para obter uma estimativa final do desempenho do modelo. Assim, Os experimentos foram realizados com três proporções de valores faltosos diferentes, sendo 10%, 20% e 30%.

Os hiperparâmetros de treinamento da rede *hyBridLSTM* (Seção 4.2.4) podem ser visualizados na Tabela 9. A rede possui 1.165.079 parâmetros, sendo 960 não treináveis. Os melhores hiperparâmetros foram encontrados utilizando o algoritmo *Tree-structured Parzen Estimator* (TPE)(BERGSTRA et al., 2011) da biblioteca *Hyperopt*¹. Além desses, a Tabela 8 apresenta também outros parâmetros da rede e são: número de blocos, *dropout*, taxa de aprendizado e épocas. Esses foram descritos na Subseção 4.2.4.

Tabela 8 – Espaço de busca dos melhores parâmetros da rede *HyBridLSTM*

Parâmetro	Espaço de busca	Selecionado
Blocos	1 a 10	6
<i>Dropout</i>	0,4 a 0,65	0,5
Taxa de aprendizado	1e-7 a 1e-2	1e-5
Épocas	100 a 2000	800
Filtros	1 a 500	112
<i>Kernel</i>	2 a 10	7
<i>Kernel regularizer</i>	L1 e L2	L1
<i>Strides</i>	2 a 5	2 e 4
<i>padding</i>	<i>Same e valid</i>	<i>Same</i>
Número de neurônios	10 a 250	32

Tabela 9 – Hiperparâmetros do modelo da rede *HybridLSTM* com melhor desempenho.

Camada	Parâmetros Aplicados				
Conv1D 1	<i>Filters =112</i>	<i>kernel_size =7</i>	<i>Kernel regularizer = L1</i>	<i>padding= same</i>	<i>strides=2</i>
Conv1D 2	<i>Filters =112</i>	<i>kernel_size =7</i>	<i>Kernel regularizer = L1</i>	<i>padding= same</i>	<i>strides=4</i>
Conv1D 3	<i>Filters =112</i>	<i>kernel_size =7</i>	<i>Kernel regularizer = L1</i>	<i>padding = same</i>	<i>strides=2</i>
Conv1D 4	<i>Filters =112</i>	<i>kernel_size =7</i>	<i>Kernel regularizer = L1</i>	<i>padding = same</i>	<i>strides=4</i>
Conv1D 5	<i>Filters =112</i>	<i>kernel_size =7</i>	<i>Kernel regularizer = L1</i>	<i>padding = same</i>	<i>strides=2</i>
Conv1D 6	<i>Filters =112</i>	<i>kernel_size =7</i>	<i>Kernel regularizer = L1</i>	<i>padding = same</i>	<i>strides=4</i>
BiLSTM	<i>units = 32</i>	<i>Dropout=0.5</i>	-	-	-
LSTM 1	<i>units = 32</i>	-	-	-	-
Dense1	<i>units = 32</i>	-	-	-	-
Dense2	units = número de classes	ativação = <i>sigmoide</i>	-	-	-
Otimizador	<i>Adam</i>	taxa de aprendizado = 1E-5	função de custo = <i>log loss</i>	-	-

Assim, inicialmente, apresentam-se os resultados dos experimentos utilizando o método de imputação proposto, usando a rede *Pix2Pix*, e 10 técnicas clássicas de imputação, totalizando 11 técnicas. Em seguida, mostram-se os resultados utilizando

¹ Disponível em: <http://hyperopt.github.io/hyperopt/>

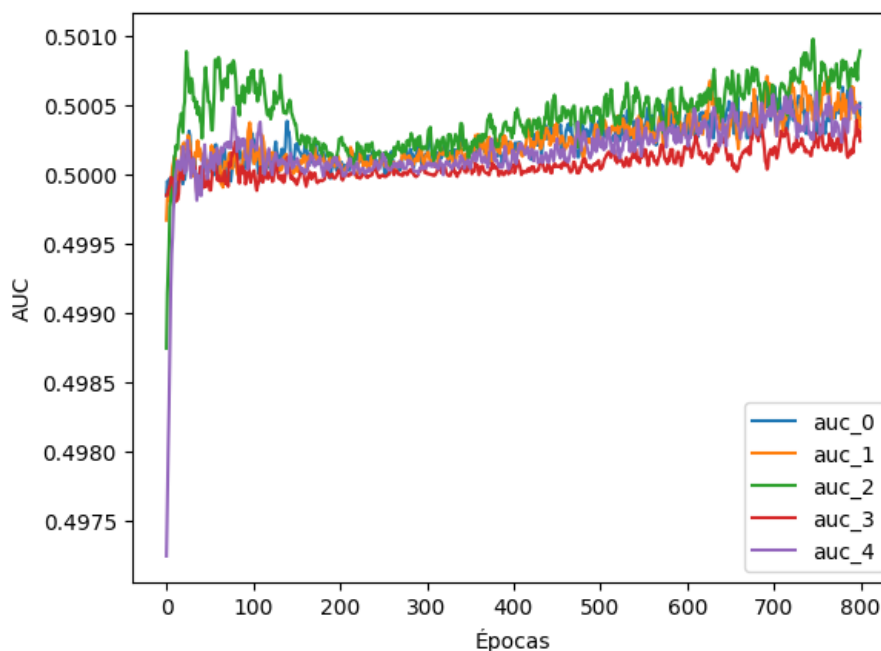
apenas as 10 técnicas clássicas de imputação. Isso torna-se necessário para que se possa ter um embasamento geral do desempenho da rede em diferentes tipos de *datasets*, pois o primeiro experimento, limita a quantidade de *datasets* utilizados.

5.3.1 Experimento com a HybridLSTM incluindo a rede Pix2Pix

Para o primeiro experimento foram utilizados 5 *datasets*, que atendiam o requisito da entrada de rede Pix2Pix que é uma entrada de 1024 elementos.

O meta-aprendizado por meio da rede foi possível meso com um desempenho mediano da rede. Muito embora tenhamos utilizado com função de custo a Área sob a curva ROC (AUC), o teste foi realizado utilizando a probabilidade máxima dos vetores de predição. Assim, na Figura 24 nota-se que ainda há possibilidade de melhoria do aprendizado, porém já com esse desempenho a rede conseguiu superar a maioria das técnicas de imputação quando tratamos série a série, e supera todas as técnicas quando avaliamos o desempenho geral. A rede consegue um resultado ligeiramente melhor na *rodada 2* e menor na *rodada 3*.

Figura 24 – AUC do treino da rede por execução Holdout no experimento com 10% de valor faltoso.



A Tabela 10 mostra a média dos resultados das primeiras 14 séries em relação à métrica ASMAPE. Além disso, consta na tabela a média e desvio padrão (SDT) dos erros de cada técnica utilizada. Por exemplo, referente a série MT_001 tem-se a média de todas as colunas nas 5 amostras igual a 0,023 a rede hybridLSTM conseguiu atingir 25,10% na

meta imputação e a mesma poderia alcançar um resultado melhor de 16,40% descrito na coluna *Objetivo*. A coluna objetivo, refere-se a melhor métrica de ASMAPE alcançada considerando a técnica que apresentou o menor ASMAPE, rotulada no metadado.

Tabela 10 – Média do ASMAPE de 5 execuções do Holdout nas primeiras 14 séries com 10% de dados faltosos

	Objetivo	M	MD	IL	IC	IA	IP5	ISP5	MM	BF	FF	P2P	Meta	STD
MT_001	16,40%	31,31%	27,56%	25,10%	32,11%	26,72%	36,89%	23,89%	24,10%	24,81%	25,07%	21,49%	25,10%	0,023
MT_002	1,81%	7,14%	6,91%	1,96%	2,61%	2,01%	3,21%	2,44%	2,59%	2,17%	2,51%	40,65%	1,96%	0,012
MT_003	2,59%	5,80%	4,89%	3,06%	4,29%	3,25%	5,31%	4,30%	3,68%	3,37%	3,42%	34,24%	3,06%	0,013
MT_006	4,16%	11,94%	11,71%	4,31%	5,15%	4,45%	6,21%	6,52%	5,21%	5,32%	5,15%	33,23%	4,31%	0,022
MT_010	3,60%	17,19%	15,13%	3,76%	3,79%	3,81%	4,26%	4,17%	5,65%	4,51%	5,10%	29,51%	3,76%	0,019
MT_011	2,93%	12,20%	12,22%	2,94%	3,28%	2,99%	3,69%	3,56%	4,44%	3,71%	4,04%	33,08%	2,94%	0,011
MT_013	3,31%	16,78%	16,18%	3,35%	3,62%	3,36%	4,17%	4,09%	5,12%	4,52%	4,70%	29,42%	3,35%	0,016
MT_019	3,44%	13,86%	13,17%	3,47%	4,02%	3,59%	4,76%	4,45%	5,04%	4,73%	4,70%	33,33%	3,47%	0,014
MT_020	3,30%	10,10%	9,97%	3,30%	3,81%	3,43%	4,52%	4,20%	4,87%	4,64%	4,41%	37,31%	3,30%	0,007
MT_021	2,73%	14,62%	14,30%	2,76%	3,32%	2,89%	4,07%	3,67%	4,00%	4,15%	3,69%	28,98%	2,76%	0,012
MT_023	3,80%	16,51%	16,69%	4,00%	4,73%	4,26%	5,55%	4,59%	5,41%	5,59%	4,41%	37,11%	4,00%	0,020
MT_025	3,24%	12,62%	12,35%	3,32%	3,80%	3,39%	4,70%	3,71%	4,53%	4,41%	4,29%	30,66%	3,32%	0,010
MT_027	3,17%	13,04%	12,74%	4,06%	5,64%	4,44%	7,17%	5,92%	3,76%	4,79%	4,37%	33,62%	4,06%	0,025
MT_029	1,63%	13,50%	12,74%	1,69%	1,85%	1,68%	2,17%	2,03%	2,72%	2,31%	2,41%	39,17%	1,69%	0,013

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF), Pix2Pix (2P2) e Desvio Padrão (STD).

Por outro lado, a Tabela 11 mostra a média geral do ASMAPE por execução do Holdout e assim confirma-se que a rede utilizada no meta-aprendizado consegue superar cada técnica quando se trata de desempenho geral.

Tabela 11 – Média do ASMAPE de 374 séries para cada execução Holdout com 10% de dados faltosos por execução Holdout

	Round 1	Round 2	Round 3	Round 4	Round 5	Média	SDT
Objetivo	6,36%	6,67%	6,94%	6,60%	6,74%	6,66%	0,002
M	30,06%	32,58%	31,00%	31,00%	31,24%	30,77%	0,009
MD	18,46%	18,73%	18,38%	18,06%	18,04%	18,33	0,003
IL	8,70%	9,32%	9,53%	9,11%	9,19%	9,17%	0,003
IC	15,64%	18,09%	17,65%	17,26%	17,14%	17,15%	0,009
IA	9,09%	9,71%	9,97%	9,50%	9,57%	9,56%	0,003
IP5	17,40%	20,04%	19,50%	19,28%	19,05%	19,05 %	0,010
ISP5	16,68%	19,30%	18,78%	18,63%	18,39%	18,35%	0,010
MM	9,46%	10,06%	10,17%	9,57%	9,86%	9,82 %	0,003
BF	8,40%	8,65%	8,87%	8,55%	8,71%	8,63 %	0,002
FF	8,35%	8,64%	9,00%	8,51%	8,65%	8,63 %	0,002
P2P	31,93%	32,02%	32,96%	33,07%	32,86%	32,56%	0,005
Meta	8,15%	8,48%	8,70%	8,38%	8,62%	8,46%	0,002

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF), Pix2Pix (2P2) e Desvio Padrão (STD).

Avaliou-se, também, a métrica MAE Tabela 12 e notou-se que na maioria das

execuções a rede hybridLSTM conseguiu ótimos resultados. Contudo, o método de imputação proposto, usando a rede Pix2Pix, conseguiu o melhor resultado, sendo também o mais estável quando se refere ao MAE. Para isso, levantaram-se algumas hipóteses. A primeira refere-se a rede, ela tem sua função de custo baseado no MAE, logo o objetivo da rede é encontrar o menor custo dessa função. Além disso, pode ter ocorrido da rede não ter considerado possíveis *outliers* nos dados, devido à normalização adotada ou ainda a transformação da série em imagem pode ter sido o diferencial.

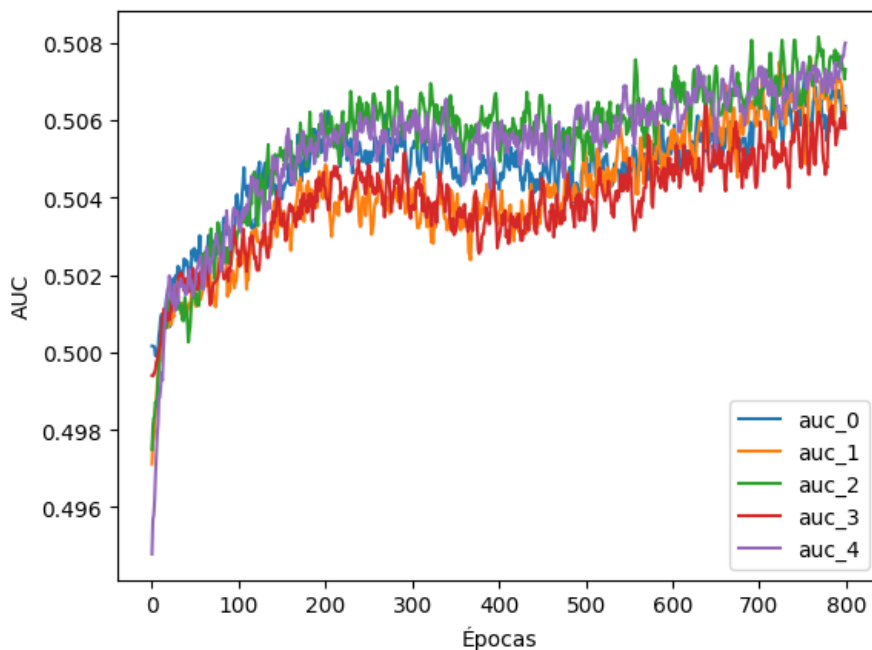
Tabela 12 – Média geral do erro MAE de 374 séries nas imputações com 10% de valor faltoso por execução Holdout

	Round 1	Round 2	Round 3	Round 4	Round 5	Média	STD
Objetivo	26,05	22,32	16,38	21,35	27,59	22,73	4,39
M	90,84	94,77	88,62	98,39	112,22	96,96	9,31
MD	94,36	104,49	89,66	93,52	111,16	98,63	8,89
IL	24,80	20,33	18,80	23,53	32,48	23,98	5,32
IC	22,98	19,19	16,61	25,32	26,58	22,13	4,18
IA	25,88	20,31	19,40	23,55	32,65	24,35	5,31
IP5	26,83	21,90	19,29	28,82	31,45	25,65	4,99
ISP5	27,56	26,40	18,65	29,78	32,05	22,08	5,09
MM	29,76	24,18	24,37	26,63	40,96	29,18	6,96
BF	33,06	26,14	19,83	32,23	35,98	29,44	6,46
FF	27,34	32,27	21,45	29,64	37,44	29,62	5,92
P2P	14,85	14,71	13,05	13,51	14,77	14,17	0,84
Meta	24,75	22,03	18,80	23,49	33,95	24,60	5,68

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF), Pix2Pix (2P2) e Desvio Padrão (STD).

Quanto ao experimento com 20% de dados faltosos tem-se na Figura 25 um desempenho semelhante ao experimento com 10% porém com a melhor curva de aprendizado. Pode-se notar ainda que a rede obteve na *rodada* 2 e 4 melhores resultados para a métrica AUC e conseqüentemente no ASMAPE, em contrapartida, teve AUCs menores nas *rodadas* 1 e 3.

Figura 25 – AUC da rede por execução Holdout no experimento com 20% de valor faltoso.



No ASMAPE por série com 20% de dados faltosos tem-se um desempenho melhor já nas primeiras séries, quando compararmos a Tabela 10 a Tabela 13. Contudo, o resultado final alcançado por meio do meta-aprendizado (Tabela 14) ainda continua melhor que cada método individualmente.

Diferente do ASMAPE, na Tabela 15 o melhor desempenho foi alcançado pelo método de imputação proposto Pix2Pix. Essa diferença se dá porque não se pode relacionar diretamente o desempenho delas. Suponham-se uma amostra $x = [2; 5; 3; 2]$, e x com valor faltoso $x = [2; nan; 3; nan]$, também duas imputações por duas técnicas quaisquer com resultados $y_1 = [2; 4, 5; 3; 1, 5]$ e $y_2 = [2; 5, 5; 3; 2, 5]$. Tem-se para o MAE

$$MAE_{y_1} = \frac{|5 - 4,5| + |2 - 1,5|}{2} = 0,5 \quad (5.1)$$

$$MAE_{y_2} = \frac{|5 - 5,5| + |2 - 2,5|}{2} = 0,5 \quad (5.2)$$

e ASMAPE

$$ASMAPE_{y_1} = \frac{\frac{|5-4,5|}{|5|+|4,5|} + \frac{|2-1,5|}{|2|+|1,5|}}{2} \simeq 0,098 \quad (5.3)$$

$$ASMAPE_{y_2} = \frac{\frac{|5-5,5|}{|5|+|5,5|} + \frac{|2-2,5|}{|2|+|2,5|}}{2} \simeq 0,103 \quad (5.4)$$

Assim, pode-se notar que ainda que duas técnicas imputem simetricamente valores faltosos, a escolha do que subestima a imputação não resultará em um MAE melhor ou pior do que outro que superestima a imputação. Isso pode ser verificado analisando as Tabelas 15 e 11.

Tabela 13 – Média do ASMAPE das 5 execuções Holdout nas primeiras 14 séries com 20% de dados faltosos

	Objetivo	M	MD	IL	IC	IA	IP5	ISP5	MM	BF	FF	P2P	Meta	STD
MT_001	18,11%	33,92%	27,97%	19,88%	26,91%	21,09%	32,53%	19,11%	19,48%	20,16%	20,01%	25,95%	19,88%	0,05
MT_003	2,57%	6,67%	5,54%	2,98%	4,13%	3,01%	5,23%	6,35%	3,23%	3,28%	3,06%	32,37%	2,98%	0,08
MT_005	11,24%	28,28%	28,06%	22,30%	22,79%	22,34%	23,54%	23,15%	23,26%	23,10%	23,19%	32,37%	22,30%	0,05
MT_006	4,22%	13,79%	13,40%	4,45%	5,83%	4,68%	7,42%	6,53%	4,94%	5,63%	5,16%	26,90%	4,45%	0,06
MT_028	6,69%	21,23%	20,95%	12,25%	12,68%	12,26%	13,29%	13,11%	13,52%	13,16%	13,25%	31,09%	12,25%	0,06
MT_032	35,74%	82,75%	82,62%	81,02%	81,41%	81,06%	81,80%	81,59%	81,00%	81,24%	81,20%	40,32%	81,02%	0,16
MT_036	29,35%	64,99%	64,94%	62,34%	63,29%	62,56%	64,29%	63,38%	62,18%	62,64%	62,66%	38,46%	62,34%	0,11
MT_041	34,49%	81,92%	81,90%	81,20%	81,61%	81,29%	82,12%	81,38%	81,16%	81,35%	81,34%	39,04%	81,20%	0,17
MT_043	15,04%	40,38%	39,50%	32,30%	32,72%	32,36%	33,21%	33,19%	33,04%	33,05%	32,90%	25,84%	32,30%	0,06
MT_044	39,07%	87,82%	85,83%	83,63%	87,14%	83,84%	87,31%	87,79%	83,52%	83,22%	82,71%	47,66%	83,63%	0,16
MT_048	13,71%	34,00%	33,86%	25,42%	27,29%	25,83%	29,38%	27,46%	24,94%	26,09%	25,90%	31,51%	25,42%	0,05
MT_050	31,28%	85,85%	72,52%	66,90%	74,68%	67,11%	76,98%	76,92%	67,37%	64,93%	66,66%	39,48%	64,94%	0,15
MT_053	12,51%	48,67%	37,51%	23,60%	24,65%	23,60%	25,63%	26,03%	25,11%	24,74%	24,33%	27,19%	23,60%	0,08
MT_063	12,52%	40,54%	39,47%	24,11%	25,91%	24,31%	27,65%	26,52%	24,78%	25,16%	24,86%	30,59%	24,11%	0,07

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), spline (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF), Pix2Pix (2P2) e Desvio Padrão (STD).

Tabela 14 – Média geral do erro ASMAPE nas imputações por execução Holdout com 20% de valores faltosos por execução Holdout.

	Round 1	Round 2	Round 3	Round 4	Round 5	Média	STD
Objetivo	9,79%	10,13%	8,94%	10,57%	9,60%	9,80 %	0,01
M	35,87%	38,52%	35,00%	37,94%	35,33%	36,732%	0,02
MD	25,23%	26,31%	25,72%	26,31%	25,89%	25,89%	0,00
IL	16,77%	17,57%	15,88%	18,06%	16,57%	16,97%	0,01
IC	24,80%	25,65%	22,03%	24,19%	22,03%	23,94%	0,02
IA	17,05%	17,83%	16,06%	18,36%	16,88%	17,23%	0,01
IP5	26,72%	27,45%	23,59%	26,08%	23,56%	25,88%	0,02
ISP5	25,67%	26,55%	22,82%	25,19%	22,87%	24,202%	0,02
MM	18,49%	18,63%	16,61%	18,71%	17,92%	16,872%	0,01
BF	16,78%	17,55%	15,97%	17,91%	16,55%	16,954%	0,01
FF	16,77%	17,36%	15,87%	17,92%	16,51%	30,934%	0,01
P2P	30,78%	31,17%	31,26%	31,94%	30,52%	30,934%	0,01
Meta	16,46%	17,18%	15,90%	17,60%	16,31%	16,90%	0,01

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), spline (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF), Pix2Pix (2P2) e Desvio Padrão (STD).

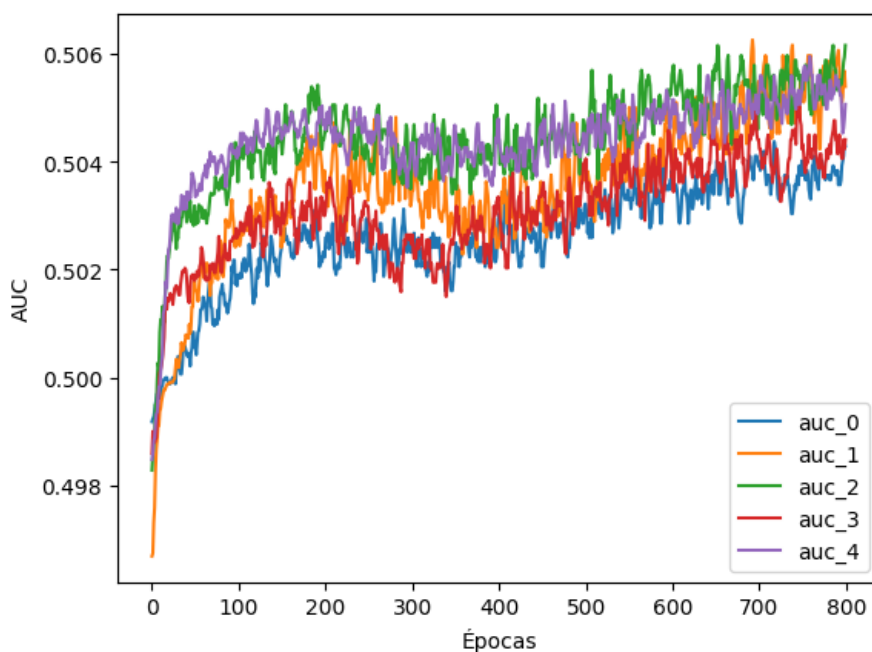
Tabela 15 – Média geral do erro MAE nas imputações com 20% de valor faltoso por execução Holdout

	<i>Round 1</i>	<i>Round 2</i>	<i>Round 3</i>	<i>Round 4</i>	<i>Round 5</i>	<i>Média</i>	<i>STD</i>
Objetivo	26,40	13,54	23,01	12,79	30,61	21,27	7,88
M	66,96	73,78	65,86	61,93	76,04	68,91	5,84
MD	67,46	70,68	72,02	58,75	75,56	68,89	6,37
IL	33,09	12,63	24,25	20,50	12,69	20,63	8,59
IC	15,74	14,23	13,29	13,33	14,34	14,18	1,00
IA	33,81	12,64	24,49	21,15	12,72	20,96	8,87
IP5	18,94	16,96	16,03	16,04	17,09	20,43	1,19
ISP5	19,19	17,41	22,00	30,32	18,64	21,51	5,20
MM	38,52	18,17	16,49	27,44	17,65	23,65	9,39
BF	18,80	17,18	26,15	16,40	33,37	22,38	7,26
FF	31,82	16,91	15,36	16,18	16,67	19,38	6,97
P2P	11,29	11,90	11,16	11,10	11,86	11,46	0,39
Meta	33,09	12,56	24,25	20,50	13,00	20,68	8,54

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quántupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF), Pix2Pix(P2P) e Desvio Padrão (STD).

Por fim, tem-se o resultado com 30% de valor faltoso. Pode-se notar que conforme aumenta a taxa de valores faltos a AUC da rede por série diminui, isso pode ser confirmado comparado o ASMAPE por série nas Tabelas 10, 13 e 16. Os desempenhos da AUC nas rodadas de experimento não apresentam diferença significativa dos experimentos anteriores, exceto pela diferença na rodada 3 que apresentou melhor resultado do que nos experimentos com 10% e 20% de valor faltoso.

Figura 26 – AUC da rede por execução Holdout no experimento com 30% de valor falto



Porém, apesar dos resultados mais baixos, com relação ao desempenho geral, a rede HyBridLSTM consegue superar cada técnica individualmente, podemos visualizar esses resultados na Tabela 17. Contudo, isso não se repete para o MAE Tabela 18, pois muito embora à rede consiga superar geralmente 10 técnicas de imputação, ela ainda não consegue superar a técnica P2P.

Tabela 16 – Média do ASMAPE em 374 séries de 5 execuções Holdout nas primeiras 14 séries com 30% de dados faltosos

	Objetivo	M	MD	IL	IC	IA	IP5	ISP5	MM	BF	FF	P2P	Meta	STD
MT_001	30,31%	57,38%	54,88%	52,98%	57,93%	53,84%	61,45%	52,39%	52,40%	53,05%	52,95%	32,93%	52,98%	0,33
MT_003	6,49%	12,86%	12,07%	12,03%	13,15%	12,04%	14,34%	12,90%	11,95%	12,22%	11,83%	39,43%	12,02%	0,18
MT_008	11,88%	31,75%	30,86%	22,54%	23,14%	22,63%	23,94%	23,60%	23,46%	23,54%	23,27%	30,90%	22,54%	0,34
MT_009	16,87%	47,87%	47,60%	32,40%	41,47%	32,39%	42,02%	41,77%	32,98%	33,24%	32,86%	35,01%	32,39%	0,32
MT_010	20,11%	47,18%	46,94%	41,38%	41,67%	41,41%	42,06%	41,81%	42,08%	41,95%	41,90%	32,93%	41,38%	0,42
MT_013	21,36%	51,53%	50,84%	42,49%	43,15%	42,61%	43,92%	43,42%	43,15%	43,41%	43,30%	33,16%	42,49%	0,41
MT_015	22,49%	47,20%	46,83%	43,54%	44,46%	43,73%	45,58%	44,66%	43,60%	44,19%	43,89%	34,87%	43,54%	0,41
MT_023	29,97%	64,84%	64,53%	62,55%	63,49%	62,73%	64,64%	63,60%	62,53%	62,80%	62,96%	38,31%	62,55%	0,40
MT_027	4,79%	11,70%	11,05%	5,25%	6,89%	5,58%	9,08%	7,01%	5,21%	6,05%	5,93%	25,83%	5,24%	0,02
MT_048	14,69%	30,40%	30,14%	25,84%	28,08%	26,30%	30,65%	28,97%	25,76%	26,51%	26,74%	30,99%	25,84%	0,33
MT_054	29,32%	65,35%	65,30%	61,14%	61,30%	61,17%	61,54%	61,52%	61,82%	61,71%	61,68%	35,50%	61,14%	0,42
MT_055	4,25%	25,97%	22,43%	4,88%	12,36%	4,84%	13,03%	14,71%	5,72%	5,34%	5,20%	28,82%	4,88%	0,11
MT_061	15,79%	50,30%	39,11%	27,62%	36,56%	27,58%	38,45%	38,42%	28,13%	27,90%	27,03%	29,24%	26,90%	0,35
MT_063	3,00%	23,29%	21,61%	3,05%	3,65%	3,10%	4,51%	3,98%	4,96%	4,50%	4,40%	19,88%	3,05%	0,03

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), spline (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF), Pix2Pix (2P2) e Desvio Padrão (STD).

Tabela 17 – Média do ASMAPE de 374 séries de 5 execuções Holdout com 30% de dados faltosos por execução Holdout

	Round 1	Round 2	Round 3	Round 4	Round 5	Média	STD
Objetivo	11,12%	10,02%	10,82%	10,63%	10,56%	10,65 %	0,00
M	38,63%	38,54%	39,20%	41,16%	36,71%	38,84%	0,02
MD	26,65%	27,12%	26,08%	26,15%	26,41%	26,44%	0,00
IL	18,67%	16,60%	18,32%	17,76%	17,28%	17,72 %	0,01
IC	26,46%	24,74%	26,83%	26,32%	25,24%	25,91 %	0,01
IA	19,00%	16,79%	18,65%	18,12%	17,50%	18,01 %	0,01
IP5	28,55%	26,30%	29,15%	28,71%	27,42%	28,22%	0,01
ISP5	27,67%	25,73%	28,37%	27,76%	26,12%	27,13%	0,01
MM	19,17%	18,13%	18,88%	18,76%	17,57%	18,50 %	0,01
BF	18,72%	16,63%	18,02%	17,71%	17,13%	17,65%	0,01
FF	18,59%	16,49%	17,80%	17,61%	17,10%	17,51 %	0,01
P2P	29,91%	29,31%	28,98%	29,26%	29,13%	29,31 %	0,00
Meta	18,13%	16,22%	17,88%	17,40%	16,93%	17,31 %	0,01

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), spline (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF), Pix2Pix (2P2) e Desvio Padrão (STD).

Tabela 18 – Média geral do erro MAE de 374 séries nas imputações com 30% de valor faltoso por execução Holdout.

	<i>Round 1</i>	<i>Round 2</i>	<i>Round 3</i>	<i>Round 4</i>	<i>Round 5</i>	<i>Média</i>	<i>STD</i>
Objetivo	27,24	21,86	13,23	23,08	9,77	19,64	7,27
M	80,29	85,45	68,05	57,81	52,97	68,71	13,97
MD	77,80	90,29	79,25	65,37	50,15	72,57	15,33
IL	26,31	20,21	11,88	10,54	7,51	15,89	7,75
IC	27,20	22,92	13,11	12,12	8,29	16,93	7,96
IA	25,79	20,26	11,76	10,59	7,53	15,79	7,58
ISP5	31,30	28,32	28,54	15,84	17,27	24,85	7,14
MM	19,15	25,54	18,36	28,07	10,78	20,98	6,77
BF	20,15	27,34	17,37	15,15	10,62	18,73	6,22
FF	28,59	24,19	16,76	24,35	10,06	20,99	7,35
P2P	9,05	8,73	8,67	9,28	9,25	9,00	0,28
Meta	14,08	20,21	11,88	10,54	7,51	12,44	4,76

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quántupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF), Pix2Pix (2P2) e Desvio Padrão (STD).

Ao final do experimento, verificou-se que a utilização de 11 técnicas de imputação resultou em excelentes resultados. Entretanto, existem perspectivas de melhoria ainda não exploradas devido à dificuldade em encontrar séries com amostras suficientes para serem adicionadas ao experimento. Essa dificuldade foi agravada pela necessidade de balancear os dados, o que gerou um número reduzido de amostras para treinamento, validação e teste, totalizando respectivamente 763, 161 e 161. A falta de dados e a conseqüente variabilidade de tipos de séries podem ter contribuído para o desempenho inferior obtido. Essa hipótese será confirmada na seção seguinte, onde há uma maior variedade de séries disponíveis.

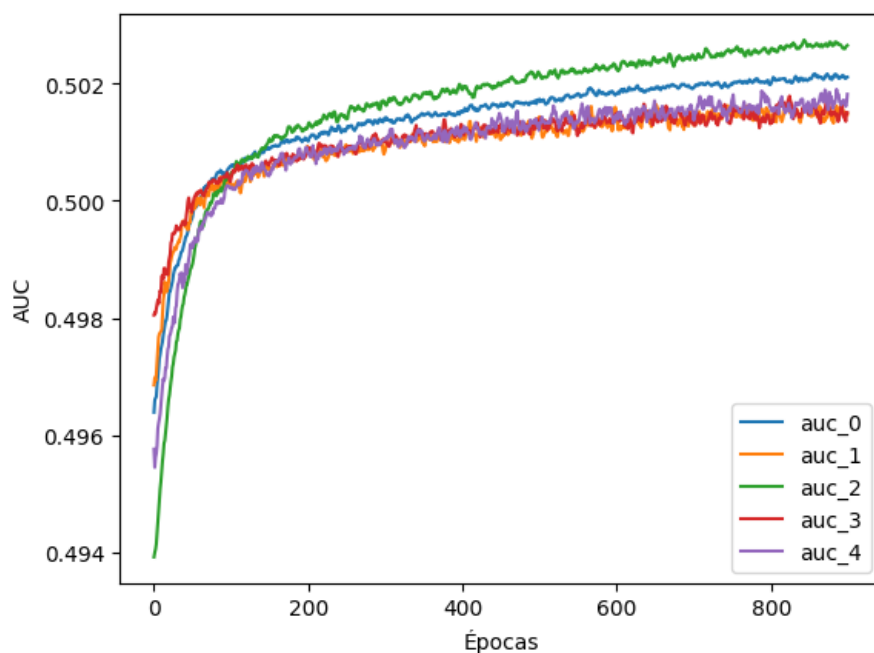
5.3.2 Experimento com a HybridLSTM sem a rede Pix2Pix

Este experimento foi realizado utilizando todas as 776 séries (Capítulo 4). Nessas séries realizou-se o janelamento com tamanho de janela 100. Após o balanceamento pela menor classe obteve-se 11.590 amostras de treino, 2.480 de validação e 2.480 de teste. Contudo, devido à natureza aleatória do experimento e pela pequena quantidade de janelas de algumas séries, não foi possível garantir a presença de todas as séries no subconjunto de teste, pois a estratificação dos dados foi realizado considerando as classes e não os *datasets*. Contudo, apresentam-se os resultados por *dataset*, uma vez que se tem ao menos três *datasets* diferentes em cada experimento.

Primeiramente, ao comparar Figura 24 com Figura 27 notam-se indícios de que a quantidade de séries de diferentes áreas impactam nos resultados, pois muito embora tenham desempenhos semelhantes em relação a AUC, a curva de aprendizado é mais suave e estável.

Diferente do experimento com 10% de dados faltosos utilizando 5 *datasets*, o experimento com 10 % de dados faltosos utilizando 14 *datasets* obteve melhor resultado na rodada 2, por outro lado, manteve um desempenho semelhante na rodada 4. De fato, não é possível garantir que as séries das amostragens nos dois experimentos relatados são as mesmas, mas nosso foco é mostrar que para o experimento em questão a diferença entre as amostragens é mínima e que houve gradativa melhora e estabilidade.

Figura 27 – AUC da rede por execução Holdout no experimento com 10% de valor faltoso em 776 séries.



A Tabela 19 mostra o resultado do ASMAPE em 4 *datasets*, pode-se observar que a rede consegue superar todas as técnicas em 50% dos *datasets* e consegue superar todos os métodos com relação à média final dos erros ASMAPE. Pode-se verificar ainda este resultado por execução Holdout na Tabela 20, na qual verifica-se que a rede HybridLSTM conseguiu atingir valores muito próximos do objetivo, ou seja, o máximo que as técnicas conseguem entregar.

Tabela 19 – Resultado ASMAPE por *Dataset* com 10% de valor faltoso.

	M	MD	IL	IC	IA	IP5	ISP5	MM	BF	FF	Objetivo	Meta
PB	20,73%	20,98%	5,09%	1,48%	2,86%	3,12%	3,85%	13,92%	7,85%	11,87%	1,47%	5,09%
EU	3,73%	3,72%	2,42%	1,47%	2,55%	2,10%	1,30%	4,92%	4,39%	3,73%	1,30%	2,02%
ELD	16,96%	13,64%	4,98%	7,25%	6,14%	8,36%	8,56%	5,67%	5,45%	5,26%	3,48%	4,06%
BSH	41,58%	40,39%	14,60%	11,28%	12,03%	14,08%	20,11%	33,92%	21,23%	26,30%	10,05%	14,60%
COVID	4,11%	4,39%	0,18%	1,20%	0,58%	1,25%	1,41%	0,34%	0,24%	0,26%	0,06%	0,24%
DMTME	12,46%	12,47%	10,81%	14,82%	11,53%	17,96%	14,63%	9,78%	18,47%	10,60%	9,78%	10,08%
SU	13,62%	13,77%	10,29%	10,83%	11,26%	18,01%	15,56%	15,50%	10,59%	17,01%	9,26%	10,29%
PBW	0,06%	0,06%	0,09%	0,14%	0,12%	0,24%	0,08%	0,07%	0,12%	0,10%	0,06%	0,09%
Média	14,16%	13,68%	6,06%	6,06%	5,89%	8,14%	8,19%	10,52%	8,54%	9,39%	4,43%	5,81%

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF) e Ffill (FF).

Nota-se ainda na Tabela 20 que o desvio padrão(STD) é muito próximo de zero, isso é reflexo da estabilidade da rede com poucos dados faltosos para imputar.

Tabela 20 – Média geral do erro ASMAPE nas imputações por execução Holdout com 10% de valores faltosos.

	Round 1	Round 2	Round 3	Round 4	Round 5	Média	STD
Objetivo	3,38%	3,34%	3,30%	3,33%	3,34%	3,34%	0,000
M	16,59%	16,30%	15,79%	16,44%	16,27%	16,28%	0,003
MD	13,29%	13,03%	13,01%	13,09%	12,97%	13,08%	0,001
IL	4,83%	4,80%	4,59%	4,84%	4,77%	4,76%	0,001
IC	7,04%	6,85%	6,42%	6,98%	6,86%	6,83%	0,002
IA	5,00%	4,97%	4,76%	4,99%	4,87%	4,92%	0,001
IP5	8,13%	7,89%	7,37%	7,94%	7,84%	7,83%	0,003
ISP5	8,23%	8,07%	7,50%	8,14%	8,03%	8,00%	0,003
MM	5,47%	5,38%	5,25%	5,50%	5,44%	5,41%	0,001
BF	5,23%	5,22%	5,12%	5,26%	5,13%	5,19%	0,001
FF	5,15%	5,11%	4,94%	5,10%	5,04%	5,07%	0,001
Meta	4,66%	4,51%	4,47%	4,51%	4,47%	4,52%	0,001

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF) e Desvio Padrão (STD).

Contudo, é importante ressaltar que o desempenho da rede HybridLSTM não se reflete de forma consistente nos resultados por *dataset*, como evidenciado na Tabela 10,

em que a rede não conseguiu identificar um número suficiente de janelas para superar as outras técnicas nos conjuntos de dados de teste. No entanto, isso não impediu que a rede HybridLSTM produzisse resultados mais estáveis e competitivos, com um erro geral próximo às técnicas de melhor desempenho e superior a todas as outras técnicas comparadas.

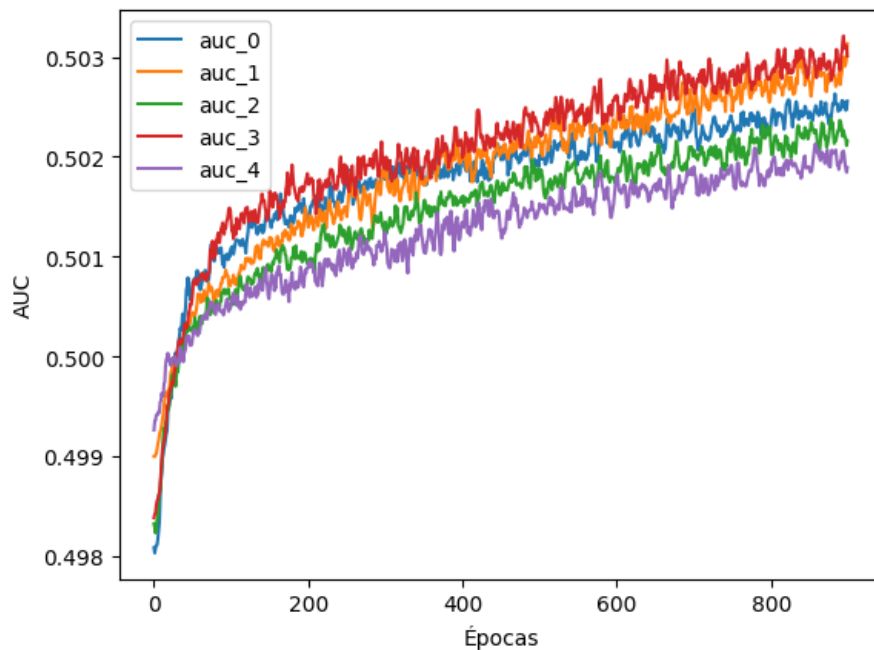
Tabela 21 – Resultado MAE por *Dataset* com 10% de valor faltoso.

Datasets	Objetivo	M	MD	IL	IC	IA	IP	ISP5	MM	BF	FF	Meta
BP	9	226,55	227,21	43,39	9,01	15,75	30,79	194,01	124,42	98,84	117,42	43,39
EU	2,54	7,18	7,16	4,59	2,83	2,93	4,14	2,54	9,49	8,7	7,19	4,59
ELD	11,88	69,61	66,61	14,08	15,56	14,1	18,3	25,19	18,4	17,8	17,56	14,14
BSH	41,22	158,68	148,63	43,19	36,73	38,98	44,08	123,89	119,98	86	93,33	43,19
DMTME	1,73	2,07	2,07	1,76	2,38	1,9	2,79	2,42	1,73	3,06	1,78	1,76
COVID	2,01	30,53	27,97	3,01	2,16	2,43	2,66	2,76	4,09	2,96	2,83	1,76
SU	3,77	5,35	5,46	4,31	4,34	4,28	5,19	4,59	4,67	5,25	4,48	4,31
PBW	0,24	0,24	0,24	0,36	0,57	0,42	1	0,32	0,29	0,51	0,42	0,36
Média	9,04	62,52	60,66	14,33	9,19	10,09	13,61	44,46	35,38	27,89	30,62	14,18

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), spline (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF) e Ffill (FF) .

Assim, quando o valor de dados faltosos é aumentado o desempenho da rede é menor, isso está visível ao compararmos as Figura 27 com a Figura 28, pois o aprendizado embora melhore bastante no *round* 4, há uma dispersão maior do que com 10% de dados faltosos.

Figura 28 – AUC da rede por execução Holdout no experimento com 20% de valor faltoso em 776 séries.



No entanto, a dispersão maior das métricas AUC nas Amostragens não impediu que a rede superasse em 33% dos *datasets*, no que diz respeito a métrica ASMAPE. Além disso, conseguiu, na média geral, superar todas as técnicas. Essa informação pode ser lida na Tabela 22. Além disso, com relação à média geral do ASMAPE por *Round* em todas as janelas mostra que a rede HyBridLSTM conseguiu manter o desempenho melhor (Tabela 23).

Tabela 22 – Resultado ASMAPE por *Datset* com 20% de valor faltoso.

	Objetivo	M	MD	IL	IC	IA	IP5	ISP5	MM	BF	FF	Meta
PB	1,31%	17,26%	15,52%	4,29%	1,53%	1,68%	2,06%	1,76%	8,45%	10,00%	7,18%	4,03%
ELD	3,80%	17,54%	13,71%	5,13%	7,73%	5,93%	8,98%	8,87%	5,70%	5,56%	5,39%	4,08%
AMM	3,44%	10,66%	11,01%	7,34%	5,09%	7,53%	3,44%	4,77%	10,17%	9,63%	8,68%	7,34%
BSH	13,15%	34,62%	35,62%	15,74%	15,57%	13,93%	22,02%	22,65%	30,62%	26,17%	26,94%	15,74%
COVID	0,09%	4,70%	4,42%	0,14%	1,16%	0,15%	1,19%	1,39%	0,27%	0,29%	0,19%	0,15%
DMTME	6,89%	9,15%	9,00%	8,65%	9,82%	8,98%	12,91%	9,51%	10,45%	8,49%	11,20%	8,15%
SU	7,56%	7,56%	7,61%	9,48%	13,47%	10,70%	16,93%	11,71%	10,24%	11,18%	9,96%	9,28%
Média	5,18%	14,50%	13,84%	7,25%	7,77%	6,99%	9,65%	8,67%	10,84%	10,19%	9,93%	6,97%

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), spline (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF) e Ffill (FF) .

Além de conseguir superar as técnicas clássicas em alguns *datasets*, a rede HyBridLSTM, também, conseguiu manter o resultado para a métrica MAE (Tabela 24), onde coincidiu de as duas métricas avaliarem os melhores métodos para imputação. Verifica-se que nos *datasets* de COVID e consumo de Energia (ELD) há janelas que aparentemente são difíceis de imputar, mas ainda assim a rede proposta conseguiu identificar em maioria dos experimentos as melhores técnicas de imputação, ou seja, conseguiu indicar as melhores técnicas de imputação na maioria dos janelamentos.

Tabela 23 – Média do ASMAPE nas Imputações por execução Holdout com 20% de valor faltoso.

	Round 1	Round 2	Round 3	Round 4	Round 5	Média	STD
Objetivo	3,67%	3,85%	3,85%	3,64%	3,70%	3,74%	0,001
M	16,78%	17,20%	17,19%	16,99%	17,18%	17,07%	0,002
MD	13,50%	13,68%	13,59%	13,55%	13,38%	13,54%	0,001
IL	4,89%	5,24%	5,18%	4,85%	5,06%	5,04%	0,002
IC	7,52%	7,88%	7,71%	7,35%	7,78%	7,64%	0,002
IA	5,05%	5,40%	5,35%	5,00%	5,25%	5,21%	0,002
IP5	8,71%	9,10%	8,97%	8,57%	8,96%	8,86%	0,002
ISP5	8,69%	8,98%	8,91%	8,51%	8,89%	8,79%	0,002
MM	5,49%	5,81%	5,76%	5,52%	5,58%	5,63%	0,001
BF	5,38%	5,67%	5,59%	5,37%	5,50%	5,50%	0,001
FF	5,21%	5,45%	5,48%	5,16%	5,29%	5,32%	0,001
Meta	4,80%	5,00%	4,92%	4,62%	4,77%	4,82%	0,001

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), spline (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF) e Desvio Padrão (STD).

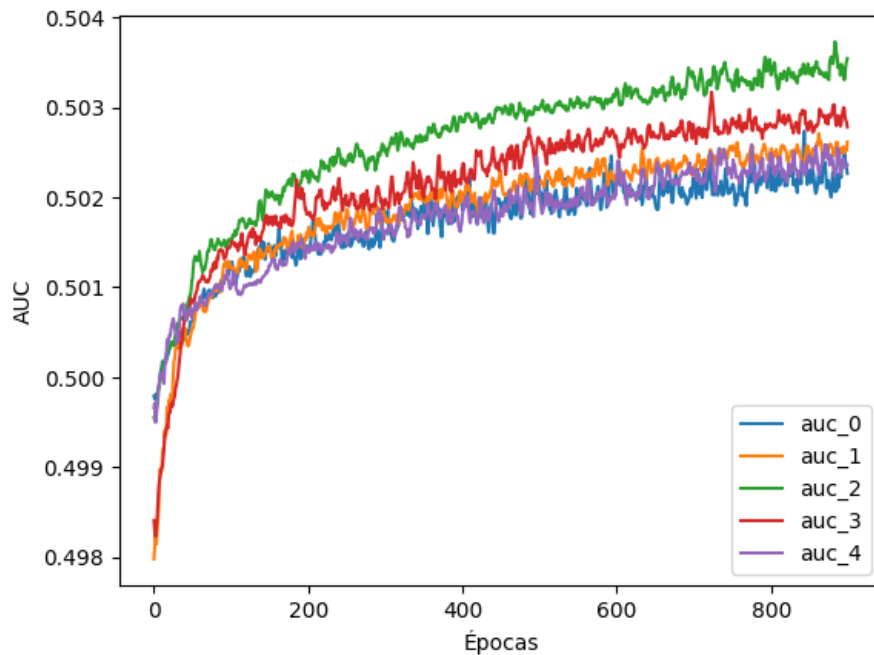
Tabela 24 – Resultado MAE Por *Datset* com 20% de valor faltoso

Datasets	Objetivo	M	MD	IL	IC	IA	IP	ISP5	MM	BF	FF	Meta
AMM	2,42	7,55	7,81	5,19	3,55	5,09	2,42	3,42	7,08	7	6,05	5,19
BP	6,26	102,62	93,03	23,78	6,93	7,61	10,45	9,76	48,84	54	40,75	23,78
ELD	13,08	70,65	67,73	14,57	16,25	14,78	19,68	23,79	18,7	18,64	17,89	14,5
BSH	44,62	123,65	123,92	49,49	46,29	45,67	58,55	59,38	102,3	82,78	93,93	49,49
COVID	5,07	29,59	32,72	3,96	2,89	3,25	5,32	4,72	9,95	8,23	7,67	5,05
DMTME	1,66	2,22	2,19	2,01	2,19	2,03	2,74	2,08	2,52	2,03	2,68	2,01
SU	1,6	1,6	1,61	1,96	2,66	2,16	3,54	2,3	2,11	2,21	2,1	1,96
Média	10,67	48,27	47,00	14,42	11,54	11,51	14,67	15,06	27,36	24,98	24,44	14,58

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF) e Ffill (FF) .

Por fim, na Figura 29 verifica-se que ao aumentar os dados faltosos de 20% para 30% AUC não há perturbação na estabilidade do aprendizado, mas consegue resultados melhores nas rodadas 2 e 3 e o desempenho permanece semelhante a Figura 28.

Figura 29 – AUC da rede por execução Holdout no experimento com 30% de valor faltoso em 776 séries.



Nesse último experimento a rede HyBridLSTM consegue manter supremacia sobre as técnicas em pelo menos dois dos maiores destes. A rede HyBridLSTM mantém o mesmo padrão de estabilidade do experimento com 10% e 20% de valor faltoso, pois a Tabela 26 mostra a mesma faixa de desvio padrão dos experimentos anteriores. Além disso, a Tabela 23 e Tabela 27 também mantém os mesmos padrões dos experimentos anteriores.

Tabela 25 – Resultado ASMAPE por *Dataset* com 30% de valor faltoso.

	Objetivo	M	MD	IL	IC	IA	IP5	ISP5	MM	BF	FF	Meta
AMM	7,26%	9,14%	9,14%	8,98%	8,24%	8,89%	7,62%	9,84%	11,67%	10,26%	10,21%	8,58%
PB	1,02%	16,12%	15,94%	4,26%	1,04%	1,98%	1,73%	1,81%	7,96%	8,98%	7,14%	3,78%
DMTME	10,48%	12,67%	12,43%	11,63%	13,44%	11,93%	15,50%	13,55%	12,57%	14,58%	12,57%	11,13%
EU	1,49%	4,21%	4,22%	3,05%	1,72%	2,74%	1,49%	1,71%	5,31%	4,37%	4,77%	3,05%
ELD	4,30%	18,23%	13,55%	5,71%	9,00%	5,94%	10,52%	10,34%	6,14%	6,11%	5,86%	5,42%
COVID	0,08%	5,27%	4,16%	0,16%	1,78%	0,15%	1,82%	2,04%	0,29%	0,30%	0,24%	0,19%
SU	11,50%	13,99%	13,97%	12,00%	14,08%	12,81%	17,83%	14,77%	14,36%	15,44%	13,75%	12,00%
MBPA	4,34%	6,68%	6,65%	4,37%	8,51%	4,78%	6,54%	8,62%	5,12%	4,34%	5,70%	4,37%
Média	4,74%	11,02%	10,13%	5,88%	7,08%	5,76%	7,92%	7,55%	7,39%	7,73%	7,14%	5,71%

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF) e Ffill (FF) .

Tabela 26 – Média do ASMAPE nas imputações por execução Holdout com 30% de valor faltoso.

	Round 1	Round 2	Round 3	Round 4	Round 5	Média	STD
Objetivo	4,01%	3,93%	4,08%	4,08%	4,15%	4,05%	0,001
M	17,68%	16,72%	17,88%	17,22%	17,62%	17,42%	0,005
MD	13,31%	13,11%	13,40%	13,45%	13,35%	13,33%	0,001
IL	5,19%	5,15%	5,52%	5,30%	5,52%	5,34%	0,002
IC	8,35%	7,88%	8,80%	8,49%	8,79%	8,46%	0,004
IA	5,38%	5,32%	5,73%	5,49%	5,74%	5,53%	0,002
IP5	9,76%	9,28%	10,25%	9,96%	10,23%	9,90%	0,004
ISP5	9,75%	9,14%	10,21%	9,84%	10,10%	9,81%	0,004
MM	5,69%	5,57%	5,89%	5,76%	5,89%	5,76%	0,001
BF	5,69%	5,70%	5,90%	5,76%	5,93%	5,80%	0,001
FF	5,43%	5,31%	5,60%	5,51%	5,69%	5,51%	0,001
Meta	5,00%	4,92%	5,20%	5,10%	5,25%	5,09%	0,001

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF) e Desvio Padrão (STD).

Tabela 27 – Resultado MAE por *Dataset* com 30% de valor faltoso.

Datasets	Objetivo	M	MD	IL	IC	IA	IP	ISP5	MM	BF	FF	Meta
AMM	4,98	6,56	6,57	6,34	5,8	6,28	5,24	7,13	8,01	7,31	7	6,34
PB	14,78	265,84	266,2	58,1	14,72	21,02	21,29	23,42	115,55	128,88	101,52	49
DMTME	2,05	2,65	2,59	2,27	2,67	2,35	3,14	2,59	2,47	3,09	2,42	2,27
EU	3,01	8,53	8,54	6,15	3,48	5,54	3,01	3,45	10,86	8,84	9,75	6,15
ELD	12,62	64,44	61,71	14,18	15,6	14,39	20,01	25,49	17,91	18,16	17,13	14,15
COVID	5,27	16,23	15,14	6,96	6,02	6,26	7,6	4,13	9,97	9,03	8,12	9,31
SU	1,91	2,37	2,36	1,95	2,14	2,07	2,81	2,3	2374	2618	2315	1,95
MBPA	10,33	15,89	15,82	10,49	16,79	11,4	15,44	36,67	12,2	10,33	13,73	10,49
Media	6,86	47,81	47,36	13,30	8,40	8,66	9,81	13,14	318,87	350,45	309,33	12,45

Média (M), Mediana (MD), Interpolações (linear (IL), cúbica (IC), slinear (IS), akima (IA), polinomial quintupla (IP5), polinomial spline grau 5 (IPS5)), média móvel (MM) Backfill(BF), Ffill (FF) e Desvio Padrão (STD).

Dado o exposto, verifica-se que os resultados mostraram que o método de imputação proposto pode competir e superar a maioria das técnicas na maioria dos conjuntos

de dados. Também mostra-se que a é mais benéfico à combinação dessas técnicas do que o usá-las separadamente, e esse benefício tornou-se possível por meio da meta-imputação, ou seja, utilizando o meta-aprendizado para imputar dados faltosos.

6 CONCLUSÃO

As séries temporais são de suma importância em diversos contextos, isso devido sua representação de processos reais, pois sua instabilidade e aleatoriedade são anotações do que é processo do dia a dia. Devido a esse fato, os estudo desse tipo de dado é difícil, porém necessário.

Quando abordamos as séries temporais com vista a completude dos dados, surge o problema de dados faltosos que se perdem por problemas como equipamento de coleta de dados quebrado ou eventos que interrompem a transferência como as catástrofes naturais. Assim, buscou-se contribuir com o estudo de imputação de dados faltosos em séries temporais univariadas.

Para isso, esse estudo primeiramente construiu uma base de dados heterogênea nas diferentes áreas de séries temporais. Selecionou-se um conjunto de técnicas de imputação de dados faltosos. Além dessas, propôs-se uma técnica de imputação de dados faltosos usado a dese Pix2Pix GAN. Seguindo, realizou-se uma subdivisão das séries temporais em janelas e, nessas janelas, imputou-se valores faltosos removidos aleatoriamente. Os métodos que melhor se saíram nas imputações foram marcados como rótulos e juntamente coma janelas constituiu o metadado. Por fim, propôs-se uma arquitetura de rede para recomendar o melhor método de imputação para cada janela. Além disso, mostrou-se como se realiza a imputação por meio de meta-aprendizado.

Os resultados desse estudo mostrou, primeiramente, que dados perdidos não são fáceis de se recuperar com altos níveis de qualidade. Observou-se que mesmo utilizando diversas técnicas de imputação os resultados não chegaram aos melhores patamares. Notou-se que o erro médio absoluto das imputações é elevado devido à complexidade das séries e presença de valores discrepantes nos dados.

Contudo, o foco principal desse trabalho esteve primeiramente em mostrar que é possível utilizar as técnicas mais simples de imputação para gerar resultados que separadamente elas não alcançariam. Para conseguir extrair o melhor da imputação de um conjunto de técnicas clássicas de imputação de dados, propôs-se a rede HybriLSTM que foi capaz, com apenas 50.3 % da métrica de avaliação AUC, sugerir técnicas para cada série que foram capazes de juntos gerar uma imputação onde o ASMAPE de cada técnica sugerida foi melhor.

Esses resultados foram possíveis porque montamos um método de construção de metadado simples, mas eficiente. Além da construção da rede HybriLSTM esse estudo também propôs um método de imputação que imputa dados ausentes por meio da transformação de séries temporais em imagens e utilização de uma rede adversária generativa, denominada Pix2Pix.

Por fim, mostrou-se que é possível utilizar várias técnicas para imputar uma mesma série localmente e não somente global. Essa abordagem ainda não foi verificada na literatura científica, por meio dela conseguimos excelentes resultados.

6.0.1 Vantagens e Limitações

Este trabalho propõe um método de imputação de dados utilizando meta-aprendizado (meta-imputação) (Seção 4.2). A meta-imputação proposta, ao contrário dos trabalhos relacionados (Capítulo 2), utiliza várias técnicas para imputar trechos específicos de uma série, e não apenas uma técnica para toda a extensão da série. Essa abordagem, mesmo que a rede o melhor valor para um determinado trecho de uma série, as outras sugestões para os outros locais da ST permitem uma maior chance de melhorar o resultado da imputação. Se apenas uma técnica fosse usada, se a rede não acertasse a técnica correta, não haveria mais a possibilidade de se obter uma imputação eficiente. Dessa forma, o método proposto reduz o risco de erro e otimiza o resultado da imputação.

Além disso, o método de transformação de STs em imagens abre espaço para uma diversidade de novos estudos que se mostrou muito eficiente e adequado para imputação via redes GANs. Mostrou-se também que a rede proposta, HybridLSTM, é capaz de reconhecer os padrões de cada ST e relacioná-lo a um tipo de técnica de imputação, havendo ainda muito espaço para resultado melhores.

Contudo, há limitações que o método proposto possui, e estes são:

1. Dificuldade de obtenção de séries temporais univariadas, pois encontrar séries univariadas com dados completos e com número de amostras maior que 1024 elementos é uma tarefa difícil;
2. Quantidade de técnicas de imputação, onde pode-se verificar que quanto maior a quantidade de técnicas de imputação maior é a concorrência entre essas e por isso melhor o resultado;

3. Tamanho das séries temporais, que devido à limitação da técnica proposta, séries com a quantidade de dados inferiores 1024 elementos não podem ser utilizadas pela rede Pix2Pix.

6.1 Contribuições

Destacam-se como principais contribuições do estudo realizado:

- Uma nova abordagem para imputação de dados faltosos em séries temporais univariadas (STU) transformando as STU em imagens e recuperando os valores faltosos por meio da rede Pix2Pix GAN.
- Um novo método de imputação de dados faltoso em STU utilizando meta-aprendizado para recomendar a melhor técnica de imputação a ser utilizada em uma janela de observações em uma ST.
- Proposição da métrica de avaliação denominada ASMAPE.

6.2 Trabalhos Futuros

Esse trabalho possui alguns caminhos que podem ser exploradas mais adiante e estes são: (1) aumentar a quantidade de séries temporais na base de dados; (2) melhorar o desempenho da rede, seja por meio de uma nova arquitetura, como blocos de rede GRU, por tratamento diferente dos dados ou aplicando as duas abordagens; (3) aumentar a gama de técnicas de imputação de dados como: imputação com aprendizado profundo, usando redes neurais profundas para prever valores faltantes em um conjunto de dados. Essa técnica tem se mostrado muito eficaz para lidar com grandes conjuntos de dados complexos, imputação baseada em grafos, usando algoritmos de grafos para modelar as interações entre as variáveis em um conjunto de dados, permitindo a imputação precisa de valores faltantes com base em informações contextuais, imputação baseada em amostragem, esta técnica usa a amostragem de Monte Carlo para gerar múltiplas versões completas do conjunto de dados, cada uma das quais é imputada com base em diferentes pressupostos sobre a distribuição dos valores faltantes. Essas técnicas contribuem para que se tenha uma rede com capacidade mais robusta; (4) explorar a imputação via janelamento com

outros tipos de rede como as GANs; (5) adaptar o método proposto para imputar dados em séries multivariadas.

6.3 Produções Científicas

Como resultado desta pesquisa, um artigo científico com o método de imputação desenvolvido foi publicado, conforme mostra a Tabela 28.

Tabela 28 – Artigo publicado que possui relação com o método proposto.

Revista	Título	Qualis
IEEE	<i>Univariate Time Series missing data Imputation using Pix2Pix GAN</i>	A4

REFERÊNCIAS

- ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JOZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANÉ, D.; MONGA, R.; MOORE, S.; MURRAY, D.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.; VANHOUCHE, V.; VASUDEVAN, V.; VIÉGAS, F.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y.; ZHENG, X. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. 2015. Software available from tensorflow.org. Disponível em: <<https://www.tensorflow.org/>>.
- AFRIFA-YAMOAHA, E.; MUELLER, U. A.; TAYLOR, S.; FISHER, A. Missing data imputation of high-resolution temporal climate time series data. **Meteorological Applications**, Wiley Online Library, v. 27, n. 1, p. e1873, 2020.
- ALHARTHI, S.; ALHARTHI, A.; ALMAATOUQ, A.; ALQAHTANI, M. Comparing the performance of machine learning algorithms in predicting type 2 diabetes using data mining techniques. **Healthcare Informatics Research**, Korean Society of Medical Informatics, v. 27, n. 4, p. 292–301, 2021.
- ALMEIDA, M. M.; ALMEIDA, J. D. S. de; JUNIOR, G. B.; SILVA, A. C.; PAIVA, A. C. de. Univariate time series missing data imputation using pix2pix gan. **IEEE Latin America Transactions**, v. 100, n. XXX, 2023.
- ARSHAM, H. Time-critical decision making for business administration. **University of Baltimore, Maryland, USA (Accessed January 2006)**, <http://home.ubalt.edu/ntsbarsh/Business-stat/stat-data/Forecast.htm>, 1994.
- ARVANITIDIS, G.; HANSEN, L. K.; HAUBERG, S. Latent space oddity: on the curvature of deep generative models. **arXiv preprint arXiv:1710.11379**, 2017.
- ASADI, R.; REGAN, A. A convolution recurrent autoencoder for spatio-temporal missing data imputation. **arXiv preprint arXiv:1904.12413**, 2019.
- ASLAN, G.; FOUMELIS, M.; RAUCOULES, D.; MICHELE, M. D.; BERNARDIE, S.; CAKIR, Z. Landslide mapping and monitoring using persistent scatterer interferometry (psi) technique in the french alps. **Remote Sensing**, MDPI, v. 12, n. 8, p. 1305, 2020.
- ATIYA, A. Why does forecast combination work so well? **International Journal of Forecasting**, v. 36, p. 197–200, 2020.
- BAHNSEN, A. C. **Australian Monthly Beer Production**. 2021. Disponível em: <<https://www.kaggle.com/code/mpwolke/australian-monthly-beer-production/data>>.
- BANDARA, K.; BERGMEIR, C.; SMYL, S. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. **Expert systems with applications**, Elsevier, v. 140, p. 112896, 2020.
- BAU, D.; ZHU, J.-Y.; STROBELT, H.; ZHOU, B.; TENENBAUM, J. B.; FREEMAN, W. T.; TORRALBA, A. Gan dissection: Visualizing and understanding generative adversarial networks. **arXiv preprint arXiv:1811.10597**, 2018.

BERGSTRA, J.; BARDENET, R.; BENGIO, Y.; KÉGL, B. Algorithms for hyperparameter optimization. **Advances in neural information processing systems**, v. 24, 2011.

BEZERRA, M. I. S. Apostila de análise de séries temporais. **UNESP: Curso de estatística. São Paulo, SP**, 2006.

BJORCK, N.; GOMES, C. P.; SELMAN, B.; WEINBERGER, K. Q. Understanding batch normalization. **Advances in neural information processing systems**, v. 31, 2018.

BOKDE, N.; BECK, M. W.; ÁLVAREZ, F. M.; KULAT, K. A novel imputation methodology for time series based on pattern sequence forecasting. **Pattern recognition letters**, Elsevier, v. 116, p. 88–96, 2018.

BRAEI, M.; WAGNER, S. Anomaly detection in univariate time-series: A survey on the state-of-the-art. **arXiv preprint arXiv:2004.00433**, 2020.

BROCK, G. N.; SHAFFER, J. R.; BLAKESLEY, R. E.; LOTZ, M. J.; TSENG, G. C. Which missing value imputation method to use in expression profiles: a comparative study and two selection schemes. **BMC bioinformatics**, BioMed Central, v. 9, n. 1, p. 1–12, 2008.

BROCKWELL, P. J.; DAVIS, R. A. **Introduction to time series and forecasting**. [S.l.]: Springer, 2002.

BROWNLEE, J. **Conjuntos de dados**. 2019. Url <https://github.com/jbrownlee/Datasets/blob/d20fcb6402ae34e653d4513b00f39257bb37ed7f/monthly-airline-passengers.csv>.

CAILLAULT, É. P.; LEFEBVRE, A.; BIGAND, A. et al. Dynamic time warping-based imputation for univariate time series data. **Pattern Recognition Letters**, Elsevier, v. 139, p. 139–147, 2020.

CHE, Z.; PURUSHOTHAM, S.; CHO, K.; SONTAG, D.; LIU, Y. Recurrent neural networks for multivariate time series with missing values. **Scientific reports**, Nature Publishing Group UK London, v. 8, n. 1, p. 6085, 2018.

CHEN, J.; DONG, H.; QIU, Y.; HE, X.; XIN, X.; CHEN, L.; LIN, G.; YANG, K. Autodebias: Learning to debias for recommendation. In: **Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval**. [S.l.: s.n.], 2021. p. 21–30.

CHOLLET, F. et al. **Keras**. [S.l.]: GitHub, 2015. <<https://github.com/keras-team/keras>>. "Acesso em 23/01/2019".

CHOU, J.-S.; TRAN, D.-S. Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders. **Energy**, Elsevier, v. 165, p. 709–726, 2018.

DATA, o. **covid-19**. 2022. Disponível em: <<https://github.com/datasets/covid-19/blob/d6d0831a4e63af87ebc6e0828f3f42edb650ae17/data/time-series-19-covid-combined.csv>>.

DATA/IMAGE, S.; BELGIUM, R. O. of; BRUSSELS. <https://www.sidc.be/silso/home>. 2021.

DICKENS, P. M.; LARSON, J. W. Classifiers for the causes of data loss using packet-loss signatures. In: IEEE. **IEEE International Symposium on Cluster Computing and the Grid, 2004. CCGrid 2004**. [S.l.], 2004. p. 212–219.

DUNN, K. **OpenMV.net Datasets**. 2018. <<https://openmv.net/>>.

DWORK, C.; FELDMAN, V.; HARDT, M.; PITASSI, T.; REINGOLD, O.; ROTH, A. The reusable holdout: Preserving validity in adaptive data analysis. **Science**, American Association for the Advancement of Science, v. 349, n. 6248, p. 636–638, 2015.

EHLERS, R. **Séries Temporais**. 2021. Ur-
l<https://sites.icmc.usp.br/ehlers/SME0808/st1.pdf>.

ELRAHMAN, S. M. A.; ABRAHAM, A. A review of class imbalance problem. **Journal of Network and Innovative Computing**, v. 1, n. 2013, p. 332–340, 2013.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. d. L. F. d. **Inteligência artificial: uma abordagem de aprendizado de máquina**. [S.l.]: LTC, 2011.

FALCÃO, H. S.; LOVATO, A. V.; SANTOS, A.; OLIVEIRA, L.; MANIÇOBA, R. H. C.; GUIMARÃES, M. A.; SANTANA, M. S. Classificação de vagas de estacionamento com utilização de rede perceptron multicamadas. **Revista de Sistemas de Informação da FSMA, Visconde de Araújo**, v. 12, p. 41–48, 2013.

FANAEE-T, H.; GAMA, J. Event labeling combining ensemble detectors and background knowledge. **Progress in Artificial Intelligence**, Springer Berlin Heidelberg, p. 1–15, 2013. ISSN 2192-6352. Disponível em: <[WebLink]>.

FANG, C.; WANG, C. Time series data imputation: A survey on deep learning approaches. **arXiv preprint arXiv:2011.11347**, 2020.

FATTAH, J.; EZZINE, L.; AMAN, Z.; MOUSSAMI, H. E.; LACHHAB, A. Forecasting of demand using arima model. **International Journal of Engineering Business Management**, SAGE Publications Sage UK: London, England, v. 10, p. 1847979018808673, 2018.

FLORES, A.; TITO, H.; SILVA, C. Local average of nearest neighbors: Univariate time series imputation. **International Journal of Advanced Computer Science and Applications**, v. 10, n. 8, p. 45–50, 2019.

FURQUIM, S. L. L. **Aeração artificial de cursos de água**. Tese (Doutorado) — Universidade de São Paulo, 1978.

GOMES, D. M. d. O. Processos auto-regressivos de coeficientes aleatórios na modelação de dados de contagem. Universidade de Évora, 2005.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.

GRAVES, J. **Análise de Séries Temporais: beersales**. 2022. Disponível em: <<https://github.com/joshgraves3/Time-Series-Analysis/blob/6cb14d75c9891fc08ccb12f9e40da84927509f/beersales.csv>>.

GUO, Y.; LIU, Y.; OERLEMANS, A.; LAO, S.; WU, S.; LEW, M. S. Deep learning for visual understanding: A review. **Neurocomputing**, Elsevier, v. 187, p. 27–48, 2016.

HAILIN, L.; MIAO, W. Fuzzy clustering based on feature weights for multivariate time series. **Knowledge-Based Systems**, Elsevier, p. 105907, 2020.

HALLAC, D.; VARE, S.; BOYD, S.; LESKOVEC, J. Toeplitz inverse covariance-based clustering of multivariate time series data. In: **Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2017. p. 215–223.

HARUTYUNYAN, H.; KHACHATRIAN, H.; KALE, D. C.; STEEG, G. V.; GALSTYAN, A. Multitask learning and benchmarking with clinical time series data. **Scientific data**, Nature Publishing Group, v. 6, n. 1, p. 1–18, 2019.

HARVEY, A. C.; SHEPHARD, N. 10 structural time series models. Elsevier, 1993.

HAYKIN, S. **Kalman filtering and neural networks**. [S.l.]: John Wiley & Sons, 2004. v. 47.

HAYKIN, S. **Neural Networks and Learning Machines**. [S.l.]: Pearson Education India, 2010.

HESSE, C. **Image-to-Image Demo: Interactive Image Translation with pix2pix-tensorflow**. 2017. Url <https://affinelayer.com/pixsrv/>.

HOSPEDALES, T.; ANTONIOU, A.; MICAELLI, P.; STORKEY, A. Meta-learning in neural networks: A survey. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 44, n. 9, p. 5149–5169, 2021.

ISOLA, P.; ZHU, J.; ZHOU, T.; EFROS, A. A. Image-to-image translation with conditional adversarial networks. **CoRR**, abs/1611.07004, 2016. Disponível em: <<http://arxiv.org/abs/1611.07004>>.

ISOLA, P.; ZHU, J.-Y.; ZHOU, T.; EFROS, A. A. Image-to-image translation with conditional adversarial networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1125–1134.

JAVED, A.; LEE, B. S.; RIZZO, D. M. A benchmark study on time series clustering. **arXiv preprint arXiv:2004.09546**, 2020.

JEON, M.; NOH, Y.; JEON, K.; LEE, S.; LEE, I. Data gap analysis of ship and maritime data using meta learning. **Applied Soft Computing**, Elsevier, v. 101, p. 107048, 2021.

JHA, B. K.; PANDE, S. Time series forecasting model for supermarket sales using fb-prophet. In: IEEE. **2021 5th International Conference on Computing Methodologies and Communication (ICCMC)**. [S.l.], 2021. p. 547–554.

JUNNINEN, H.; NISKA, H.; TUPPURAINEN, K.; RUUSKANEN, J.; KOLEHMAINEN, M. Methods for imputation of missing values in air quality data sets. **Atmospheric environment**, Elsevier, v. 38, n. 18, p. 2895–2907, 2004.

JúnIOR, R. O. d. C. **Comparação de métodos de imputação para séries temporais**. 2022. Url <https://rpubs.com/rubensocj/imputation-univariate>.

- KANG, E. **Long Short-Term Memory (LSTM): Concept**. Medium, 2017. Disponível em: <<https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359>>.
- KANKANA. **Daily minimum temperatures in ME**. 2022. Disponível em: <<https://www.kaggle.com/datasets/kankanashukla/daily-minimum-temperatures-in-me>>.
- KARIM, F.; MAJUMDAR, S.; DARABI, H.; HARFORD, S. Multivariate lstm-fcns for time series classification. **Neural Networks**, Elsevier, v. 116, p. 237–245, 2019.
- KARPATHY, A.; FEI-FEI, L. Deep visual-semantic alignments for generating image descriptions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 3128–3137.
- KHAYATI, M.; LERNER, A.; TYMCHENKO, Z.; CUDRÉ-MAUROUX, P. Mind the gap: an experimental evaluation of imputation of missing values techniques in time series. In: **Proceedings of the VLDB Endowment**. [S.l.: s.n.], 2020. v. 13, n. 5, p. 768–782.
- KRAMER, O. Scikit-learn. In: **Machine learning for evolution strategies**. [S.l.]: Springer, 2016. p. 45–53.
- KREINOVICH, V.; NGUYEN, H. T.; OUNCHAROEN, R. How to estimate forecasting quality: A system-motivated derivation of symmetric mean absolute percentage error (smape) and other similar characteristics. 2014.
- LIAO, T. W. Clustering of time series data—a survey. **Pattern recognition**, Elsevier, v. 38, n. 11, p. 1857–1874, 2005.
- MA, C.; FENG, L. The research on the bilging and alarm control system. In: **2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)**. [S.l.: s.n.], 2017. v. 2, p. 328–331.
- MACKINNON, J. G. **Critical Values For Cointegration Tests**. [S.l.], 2010. Disponível em: <<https://ideas.repec.org/p/qed/wpaper/1227.html>>.
- MAHALAKSHMI, G.; SRIDEVI, S.; RAJARAM, S. A survey on forecasting of time series data. In: IEEE. **2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)**. [S.l.], 2016. p. 1–8.
- MARREIROS, A. C.; DAUNIZEAU, J.; KIEBEL, S. J.; FRISTON, K. J. Population dynamics: variance and the sigmoid activation function. **Neuroimage**, Elsevier, v. 42, n. 1, p. 147–157, 2008.
- MARTÍNEZ-ÁLVAREZ, F.; TRONCOSO, A.; ASECIO-CORTÉS, G.; RIQUELME, J. C. A survey on data mining techniques applied to electricity-related time series forecasting. **Energies**, MDPI, v. 8, n. 11, p. 13162–13193, 2015.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.
- MCKINNEY, W. et al. pandas: a foundational python library for data analysis and statistics. **Python for high performance and scientific computing**, Seattle, v. 14, n. 9, p. 1–9, 2011.

- MENEZES, I. B.; DUARTE, M. A. Q. Métodos de interpolação polinomial. **AnAis do enic**, n. 2, 2010.
- MESQUITA, R. R. et al. Fórmulas variacionais tipo hadamard para os autovalores do eta-laplaciano e aplicações. Universidade Federal do Amazonas-Universidade Federal do Pará, 2014.
- MLODINOW, L. **O andar do bêbado: como o acaso determina nossas vidas**. [S.l.]: Editora Schwarcz-Companhia das Letras, 2009.
- MOAYEDI, H. Z.; MASNADI-SHIRAZI, M. A. Arima model for network traffic prediction and anomaly detection. **2008 International Symposium on Information Technology**, v. 4, p. 1–6, 2008.
- MORETTIN, P. A.; TOLOI, C. M. **Análise de séries temporais: modelos lineares univariados**. [S.l.]: Editora Blucher, 2018.
- NAJAFABADI, M. M.; VILLANUSTRE, F.; KHOSHGOFTAAR, T. M.; SELIYA, N.; WALD, R.; MUHAREMAGIC, E. Deep learning applications and challenges in big data analytics. **Journal of big data**, Springer, v. 2, n. 1, p. 1–21, 2015.
- PAN, Z.; YU, W.; YI, X.; KHAN, A.; YUAN, F.; ZHENG, Y. Progresso recente em redes generativas de adversários (gans): uma pesquisa. **IEEE**, v. 7, p. 36322–36333, 2019.
- PAPARRIZOS, J.; GRAVANO, L. Fast and accurate time-series clustering. **ACM Transactions on Database Systems (TODS)**, ACM New York, NY, USA, v. 42, n. 2, p. 1–49, 2017.
- PATTON, A. J. A review of copula models for economic time series. **Journal of Multivariate Analysis**, Elsevier, v. 110, p. 4–18, 2012.
- PONTI, M. A.; COSTA, G. B. P. D. Como funciona o deep learning. **arXiv preprint arXiv:1806.07908**, 2018.
- REIS, M. M. **4 - ANÁLISE DE SÉRIES TEMPORAIS**. 2015. Ur-
l<https://www.inf.ufsc.br/marcelo.menezes.reis/Cap4.pdf>.
- RONNEBERGER, O.; FISCHER, P.; BROX, T. **U-Net: Convolutional Networks for Biomedical Image Segmentation**. arXiv, 2015. Disponível em: <<https://arxiv.org/abs/1505.04597>>.
- RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **International Conference on Medical image computing and computer-assisted intervention**. [S.l.], 2015. p. 234–241.
- ROSSUM, G. van; AL. et. **datetime — Basic date and time types**. [S.l.], 2021.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.

SEPULVEDA, L. F.; DINIZ, P. S.; DINIZ, J. O.; NETTO, S. M.; CIPRIANO, C. L.; ARAÚJO, A. C.; LEMOS, V. H.; PESSOA, A. C.; QUINTANILHA, D. B.; ALMEIDA, J. D. et al. Forecasting of individual electricity consumption using optimized gradient boosting regression with modified particle swarm optimization. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 105, p. 104440, 2021.

SHAH, V.; DIMITROV, S. A comparative study of univariate time-series methods for sales forecasting. **International Journal of Business and Data Analytics**, Inderscience Publishers (IEL), v. 2, n. 2, p. 187–216, 2022.

SHINDE, P. P.; SHAH, S. A review of machine learning and deep learning applications. In: IEEE. **2018 Fourth international conference on computing communication control and automation (ICCUBEA)**. [S.l.], 2018. p. 1–6.

STAUEMEYER, R. C.; MORRIS, E. R. Understanding lstm—a tutorial into long short-term memory recurrent neural networks. **arXiv preprint arXiv:1909.09586**, 2019.

STOLFI, D. H.; ALBA, E.; YAO, X. Predicting car park occupancy rates in smart cities. In: **Smart Cities**. Springer International Publishing, 2017. p. 107–117. Disponível em: <https://doi.org/10.1007/978-3-319-59513-9_11>.

THOMAS, T.; RAJABI, E. A systematic review of machine learning-based missing value imputation techniques. **Data Technologies and Applications**, Emerald Publishing Limited, v. 55, n. 4, p. 558–585, 2021.

TORRE, J. de la. Redes generativas adversarias (gan) fundamentos teóricos y aplicaciones. **arXiv preprint arXiv:2302.09346**, 2023.

TRINDADE, A. **ElectricityLoadDiagrams20112014 Data Set**. 2021. Disponível em: <<https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014#>>.

US, B. of Governors of the F. R. S. **Industrial production: Utilities: Electric and Gas Utilities (NAICS = 2211,2)**. 2022. Disponível em: <<https://fred.stlouisfed.org/series/IPG2211A2N>>.

VANSCHOREN, J. Meta-learning: A survey. **arXiv preprint arXiv:1810.03548**, 2018.

VANSCHOREN, J. Meta-learning. In: **Automated machine learning**. [S.l.]: Springer, Cham, 2019. p. 35–61.

WANG, Z.; YAN, W.; OATES, T. Time series classification from scratch with deep neural networks: A strong baseline. **CoRR**, abs/1611.06455, 2016. Disponível em: <<http://arxiv.org/abs/1611.06455>>.

WOLD, S. Spline functions in data analysis. **Technometrics**, Taylor & Francis, v. 16, n. 1, p. 1–11, 1974.

WU, X.; MATTINGLY, S.; MIRJAFARI, S.; HUANG, C.; CHAWLA, N. V. Personalized imputation on wearable-sensory time series via knowledge transfer. In: **Proceedings of the 29th ACM International Conference on Information & Knowledge Management**. [S.l.: s.n.], 2020. p. 1625–1634.

YANG, X.; ZHANG, Z. A cnn-lstm model based on a meta-learning algorithm to predict groundwater level in the middle and lower reaches of the heihe river, china. **Water**, MDPI, v. 14, n. 15, p. 2377, 2022.

YOZGATLIGIL, C.; ASLAN, S.; IYIGUN, C.; BATMAZ, I. Comparison of missing value imputation methods in time series: the case of turkish meteorological data. **Theoretical and applied climatology**, Springer, v. 112, p. 143–167, 2013.

ZADEH, P. M.; SAYADI, M.; KOSARI, A. An efficient metamodel-based multi-objective multidisciplinary design optimization framework. **Applied Soft Computing**, v. 74, p. 760–782, 2019. ISSN 1568-4946. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1568494618305313>>.

ZEROUAL, A.; HARROU, F.; DAIRI, A.; SUN, Y. Deep learning methods for forecasting covid-19 time-series data: A comparative study. **Chaos, Solitons & Fractals**, Elsevier, v. 140, p. 110121, 2020.

ZHUANG, Y.; KE, R.; WANG, Y. Innovative method for traffic data imputation based on convolutional neural network. **IET Intelligent Transport Systems**, IET, v. 13, n. 4, p. 605–613, 2019.