UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Instituto de Ciência e Tecnologia de Sorocaba

LUCAS ROBERTO TOMAZINI

Detecção e Localização de Vazamentos em Redes de Distribuição de Água Utilizando Aprendizado de Máquina em Estruturas de Grafo

Sorocaba
2026

LUCAS ROBERTO TOMAZINI

Detecção e Localização de Vazamentos em Redes de Distribuição de Água Utilizando Aprendizado de Máquina em Estruturas de Grafo

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Estadual Paulista "Júlio de Mesquita Filho" (Unesp) para obtenção do título de mestre em Engenharia Elétrica.

Área de Concentração: Automação
Linha de Pesquisa: Sistemas Mecatrônicos
Orientador: Prof. Dr. Alexandre da Silva Simões

Sorocaba
2026

T655d

Tomazini, Lucas Roberto
Detecção e localização de vazamentos em redes de distribuição de água utilizando aprendizado de máquina em estruturas de grafo / Lucas Roberto Tomazini. -- Sorocaba, 2026
89 p. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Alexandre da Silva Simões

1. Agua Distribuição. 2. Água Vazamento. 3. Inteligencia artificial. 4. Deep learning (Machine learning). 5. Internet das coisas. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Sorocaba

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Detecção e Localização de Vazamentos em Redes de Distribuição de Água Utilizando Aprendizado de Máquina em Estruturas de Grafo

AUTOR: LUCAS ROBERTO TOMAZINI
ORIENTADOR: ALEXANDRE DA SILVA SIMÕES

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica, área: Automação pela Comissão Examinadora:

Prof. Dr. ALEXANDRE DA SILVA SIMÕES (Participaçao Presencial)
Departamento de Engenharia de Controle e Automação / UNESP / Câmpus de Sorocaba - ICTS

Professor Titular REINALDO AUGUSTO DA COSTA BIANCHI (Participaçao Virtual)

Prof. Dr. EDUARDO PACIÊNCIA GODOY (Participaçao Virtual)
Departamento de Engenharia de Controle e Automação / UNESP / Câmpus de Sorocaba - ICTS

Sorocaba, 29 de maio de 2026.

Agradecimentos

À minha esposa Camilla, pelo apoio constante e pela paciência em todas as fases deste projeto.

Aos meus pais, Roberto e Fátima, pela educação sólida e pelo ensinamento, desde a infância, sobre a importância da busca pelo conhecimento.

Ao meu orientador, Prof. Dr. Alexandre da Silva Simões, pelo suporte incondicional, orientação precisa e confiança ao longo de todas as etapas deste trabalho.

Aos membros da banca examinadora, cujas contribuições valiosas e sugestões de melhoria enriqueceram este projeto.

Ao Rodrigo Rolle e ao Weliton Rodrigues pela parceria e troca de experiências que foram essenciais para o desenvolvimento desta pesquisa.

Ao corpo técnico da Unesp Sorocaba e a todos aqueles que direta e indiretamente contribuíram para a realização deste trabalho.

Nós fazemos o nosso mundo significativo pela coragem de nossas perguntas,
e pela profundidade de nossas respostas.
(Carl Sagan, Cosmos)

Prefácio

Este trabalho foi desenvolvido no contexto de uma parceria entre a Universidade Estadual Paulista **(Unesp)** e a empresa **Yamatec** Indústria de Ferramentas Eletromecânicas LTDA-ME com o objetivo de pesquisar e desenvolver soluções para a identificação e localização de vazamentos em redes de distribuição de água.

A Yamatec atua no segmento de geofones, equipamentos utilizados na detecção de vazamentos por meio da análise dos sons produzidos pelas falhas existentes nas tubulações. Além do desenvolvimento desses equipamentos, a empresa também presta serviços de monitoramento de redes de distribuição de água, gerando um volume significativo de dados que podem ser explorados para aprimorar os processos de detecção e localização de vazamentos.

Nesse cenário, o presente trabalho busca investigar a aplicação de técnicas de aprendizado de máquina para a análise dos dados provenientes desses sistemas de monitoramento buscando aumentar a eficiência na identificação e localização de vazamentos.

O presente trabalho integra um conjunto mais amplo de iniciativas de pesquisa e desenvolvimento conduzidas no âmbito dessa parceria, incluindo estudos sobre a aplicação de diferentes arquiteturas de redes neurais para análise de vazamentos, o desenvolvimento de um painel de monitoramento (*dashboard*) para apoio à operação dos sistemas e a concepção de um dispositivo de aquisição de dados com a capacidade de transmissão dos dados por meio de redes sem fio.

Os resultados obtidos também contribuíram para um depósito de patente realizado junto ao Instituto Nacional da Propriedade Industrial (INPI), com o apoio da Agência Unesp de Inovação (AUIN), evidenciando o potencial de aplicação prática e de inovação tecnológica das soluções desenvolvidas.

Resumo

As Redes de Distribuição de Água (RDAs) desempenham um papel essencial no abastecimento eficiente de consumidores finais. Contudo, vazamentos nessas infraestruturas resultam em significativos desperdícios hídricos e perdas financeiras. Diante desse cenário, este trabalho tem como objetivo desenvolver e validar uma metodologia baseada em Aprendizado de Máquina (AM) em estruturas de grafo para a identificação e localização de vazamentos em RDAs. O estudo foi conduzido por meio da modelagem e simulação de cenários operacionais da RDA de Hanói, utilizando a integração das ferramentas EPANET e MATLAB. As simulações contemplaram condições normais de operação e sob a incidência de vazamentos, variando sua localização, intensidade e considerando diferentes padrões de consumo. Os dados gerados foram submetidos a uma etapa de pré-processamento baseada em cálculo de resíduos e utilizados para treinar e avaliar 5 diferentes arquiteturas: *Gated Graph Neural Network* (GGNN), *Graph Long Short-Term Memory* (G-LSTM) e GraphSAGE (com agregadores Média, Pooling e LSTM). Os experimentos foram conduzidos para analisar os modelos considerando cinco aspectos principais: a avaliação do desempenho em condições ideais, a medição da degradação sob falha progressiva de múltiplos sensores, o mapeamento da importância individual de cada sensor, a identificação dos mais críticos para o aprendizado da rede e a definição do número mínimo de sensores necessários para um treinamento satisfatório. Os resultados demonstram que os modelos GraphSAGE-LSTM e G-LSTM têm ótimo desempenho em condições ideais, contudo, destaca-se a GGNN devido à sua robustez a falhas operacionais.

Palavras-chave: Redes de Distribuição de Água, Detecção de Vazamentos, Aprendizado de Máquina, Redes Neurais para Grafos, EPANET.

Abstract

Water distribution networks (WDNs) play an essential role in the efficient supply to end consumers; however, leaks in these infrastructures result in significant water waste and financial losses. Given this scenario, this work aims to develop and validate a methodology based on the use of Machine Learning (ML) in graph structures for the identification and location of leaks in WDNs. The study was conducted through the modelling and simulation of operational scenarios of the Hanoi WDN, using the integration of EPANET and MATLAB tools. The simulations included normal operating conditions and scenarios with the incidence of leaks, varying their location and intensity, and considering different consumption patterns. The generated data were submitted to a preprocessing stage based on residual calculation and used to train and evaluate 5 different architectures: GGNN, G-LSTM, and GraphSAGE (with Mean, Pooling, and LSTM aggregators). Experiments were conducted to analyze the network's performance under ideal conditions, the degradation due to the progressive failure of multiple sensors, to map the individual importance of the sensors, to discover which sensors are most important for training, and to establish the minimum number of sensors required for effective training. The results demonstrate that the GraphSAGE-LSTM and G-LSTM models show great performance under ideal conditions; however, the GGNN stands out due to its robustness to operational failures.

Keywords: Water Distribution Networks, Leak Detection, Machine Learning, Graph Neural Networks, EPANET.

Lista de ilustrações

Figura 1 – Índices de perdas na distribuição de água no Brasil. 12
Figura 2 – Invariância da permutação em grafos. 17
Figura 3 – Processo de *embedding* das GNN. 18
Figura 4 – Mecanismo de *message passing*. 19
Figura 5 – Aqueduto Pont du Gard, na França, patrimônio mundial da UNESCO. . . 25
Figura 6 – Demonstração dos métodos passivos e ativos para identificação de vazamentos em RDAs. 28
Figura 7 – Esquemático detalhado das etapas do projeto. 39
Figura 8 – Representação de uma RDA desenhada no EPANET. 43
Figura 9 – Padrão de consumo para um período de 24 horas. 45
Figura 10 – RDA da cidade de Hanói. 46
Figura 11 – Exemplo de uma matriz de confusão. 53
Figura 12 – Pontos de simulação de vazamentos. 55
Figura 13 – Padrões de consumo. 56
Figura 14 – Comparativo do F1-score dos modelos treinados com os dados de todos os sensores. 63
Figura 15 – Matriz de confusão normalizada para os modelos avaliados. 65
Figura 16 – Degradação por Falha dos Sensores para o Modelo GGNN. 66
Figura 17 – Degradação por Falha dos Sensores para o Modelo G-LSTM. 67
Figura 18 – Degradação por Falha dos Sensores para o Modelo GraphSAGE-LSTM. . . 68
Figura 19 – Degradação por Falha dos Sensores para o GraphSAGE-Média. 68
Figura 20 – Degradação por Falha dos Sensores para o Modelo GraphSAGE-Pooling. . 69
Figura 21 – Importância dos sensores para o Modelo GGNN. 70
Figura 22 – Importância dos sensores para o Modelo G-LSTM. 71
Figura 23 – Importância dos sensores para o Modelo GraphSAGE-LSTM. 71
Figura 24 – Importância dos sensores para o Modelo GraphSAGE-Média. 72
Figura 25 – Mapa de Calor da Importância dos sensores. 72
Figura 26 – Sensores críticos para o modelo GGNN. 73
Figura 27 – Sensores Críticos para o modelo G-LSTM. 74
Figura 28 – Sensores Críticos para o modelo GraphSAGE-LSTM. 75
Figura 29 – Sensores Críticos para o modelo GraphSAGE-Média. 76
Figura 30 – Mapa de Calor. 76
Figura 31 – Determinação do número mínimo de sensores para o GGNN. 77
Figura 32 – Determinação do número mínimo de sensores para o G-LSTM. 78
Figura 33 – Determinação do número mínimo de sensores para o GraphSAGE-LSTM. . 79
Figura 34 – Determinação do número mínimo de sensores para o GraphSAGE-Média. . 79

Lista de abreviaturas e siglas

BPTT	*Backpropagation Through Time* (Retropropagação no tempo)
CNN	*Convolutional Neural Networks* (Redes Neurais Convolucionais)
EPA	Agência de Proteção Ambiental dos Estados Unidos
GCN	*Graph Convolutional Network*
GGNN	*Gated Graph Neural Network*
GGSNN	*Gated Graph Sequential Neural Network*
G-LSTM	*Graph Neural Network LSTM*
GNN	*Graph Neural Network*
GraphSAGE	*Graph Sample and Aggregate*
GRU	*Gated Recurrent Unit*
IoT	*Internet of Things*
LDA	*Linear Discriminant Analysis*
LPWAN	*Low Power Wide Area Network*
LSTM	*Long Short-Term Memory*
ML	*Machine Learning*
MLP	*Multi-Layer Perceptron*
MSE	Erro Quadrático Médio
NPW	Onda de pressão negativa
PCA	*Principal Component Analysis*
RDA/RDAs	Rede(s) de Distribuição de Água
RNA	Redes Neurais Artificiais
RNF	Reservatório de Nível Fixo
RNV	Reservatório de Nível Variável

RNN	*Recurrent Neural Network*
SNIS	Sistema Nacional de Informações sobre Saneamento
SVM	*Support Vector Machine*
WDN	*Water Distribution Networks*

Sumário

1 INTRODUÇÃO 11
1.1 PROBLEMA DE PESQUISA 11
1.2 OBJETIVO 13
1.3 JUSTIFICATIVA 13
1.4 ESTRUTURA E CONTEÚDO 14

2 FUNDAMENTAÇÃO TEÓRICA 16
2.1 REDES NEURAIS PARA GRAFOS (GNN) 16
2.1.1 *LONG SHORT-TERM MEMORY GRAPH NEURAL NETWORK (G-LSTM)* 20
2.1.2 *GATED GRAPH NEURAL NETWORK (GGNN)* 21
2.1.3 *GRAPHSAGE* 23
2.2 DETECÇÃO DE VAZAMENTOS EM RDAs 24
2.2.1 REDES DE DISTRIBUIÇÃO DE ÁGUA (RDAs) 26
2.2.1.1 PERDAS NAS RDAs 26
2.2.2 MÉTODOS PARA DETECÇÃO DE VAZAMENTOS EM RDAs 27

3 TRABALHOS CORRELATOS 30
3.1 TRABALHOS QUE UTILIZAM A RDA DA CIDADE DE HANÓI . 33

4 MATERIAIS E MÉTODOS 38
4.1 CONSTRUÇÃO DOS MODELOS DE RDAs 41
4.1.1 MODELAGEM DO CONSUMO DE ÁGUA 43
4.1.2 MODELAGEM DOS VAZAMENTOS DE ÁGUA 45
4.2 RDA DA CIDADE DE HANÓI NO VIETNÃ 46
4.2.1 CÁLCULO DOS RESÍDUOS 47
4.2.2 PRÉ-PROCESSAMENTO DOS DADOS 47
4.3 IMPLEMENTAÇÃO DAS REDES NEURAIS 49
4.3.1 GGNN 49
4.3.2 G-LSTM 50
4.3.3 GraphSAGE 51
4.4 MÉTRICAS DE AVALIAÇÃO 52
4.5 PROCEDIMENTO EXPERIMENTAL 54
4.5.1 SIMULAÇÕES NO EPANET-MATLAB 54
4.5.2 TREINAMENTO, VALIDAÇÃO E TESTE 56
4.6 EXPERIMENTO 1 - ANÁLISE SOB CONDIÇÃO IDEAL 57

4.7 EXPERIMENTO 2 - ANÁLISE SOB CONDIÇÃO DE FALHA SEQUENCIAL . . . 58
4.8 EXPERIMENTO 3 - AVALIAÇÃO DA IMPORTÂNCIA DOS SENSORES NA SITUAÇÃO DE FALHA OPERACIONAL . . . 59
4.9 EXPERIMENTO 4 - AVALIAÇÃO DOS SENSORES CRÍTICOS PARA O APRENDIZADO DO MODELO . . . 60
4.10 EXPERIMENTO 5 - DETERMINAÇÃO DO NÚMERO MÍNIMO DE SENSORES PARA TREINAMENTO . . . 61

5 RESULTADOS E DISCUSSÕES . . . 63
5.1 EXPERIMENTO 1 - ANÁLISE SOB CONDIÇÃO IDEAL . . . 63
5.1.1 ANÁLISE GLOBAL DOS RESULTADOS . . . 63
5.1.2 INFLUÊNCIA DA LOCALIZAÇÃO DOS VAZAMENTOS . . . 64
5.2 EXPERIMENTO 2 - ANÁLISE SOB CONDIÇÃO DE FALHA SEQUENCIAL . . . 66
5.2.1 GGNN . . . 66
5.2.2 G-LSTM . . . 67
5.2.3 GraphSAGE-LSTM . . . 67
5.2.4 GraphSAGE-Média . . . 68
5.2.5 GraphSAGE-Pooling . . . 69
5.3 EXPERIMENTO 3 - AVALIAÇÃO DA IMPORTÂNCIA DOS SENSORES NA SITUAÇÃO DE FALHA OPERACIONAL . . . 70
5.3.1 GGNN . . . 70
5.3.2 G-LSTM . . . 71
5.3.3 GraphSAGE-LSTM . . . 71
5.3.4 GraphSAGE-Média . . . 72
5.3.5 MAPA DE CALOR . . . 72
5.4 EXPERIMENTO 4 - AVALIAÇÃO DOS SENSORES CRÍTICOS PARA O APRENDIZADO DO MODELO . . . 73
5.4.1 GGNN . . . 73
5.4.2 G-LSTM . . . 74
5.4.3 GraphSAGE-LSTM . . . 75
5.4.4 GraphSAGE-Média . . . 75
5.4.5 MAPA DE CALOR . . . 75
5.5 EXPERIMENTO 5 - DETERMINAÇÃO DO NÚMERO MÍNIMO DE SENSORES PARA TREINAMENTO . . . 77
5.5.1 GGNN . . . 77
5.5.2 G-LSTM . . . 77
5.5.3 GraphSAGE-LSTM . . . 78
5.5.4 GraphSAGE-Média . . . 78

5.6 CONSIDERAÇÕES GERAIS SOBRE OS EXPERIMENTOS ANALISADOS 80

6 CONCLUSÕES 82

REFERÊNCIAS 84

1 INTRODUÇÃO

1.1 PROBLEMA DE PESQUISA

A água é um recurso natural indispensável tanto para a vida quanto para a maioria dos processos industriais. A distribuição eficiente e sustentável de água é um dos desafios enfrentados pelas sociedades modernas. Redes de distribuição de água (RDAs) desempenham um papel fundamental na garantia do fornecimento de água potável aos consumidores finais, abrangendo áreas residenciais, comerciais, instalações industriais e campus universitários. No entanto, vazamentos ao longo dessas redes representam um problema significativo, resultando em desperdício de água, perdas financeiras e impactos negativos ao meio ambiente.

Os vazamentos em RDAs são uma preocupação global, afetando países em diferentes estágios de desenvolvimento. Entretanto, como salientado por Nkemeni, Mieyeville e Tsafack (2020), tanto as consequências ambientais quanto as econômicas são mais significativas nos países em desenvolvimento, onde os prejuízos financeiros decorrentes de vazamentos ultrapassam 3 bilhões de dólares por ano.

Cerca de um terço dos países do mundo apresenta perdas nas RDAs decorrentes de vazamentos não identificados superiores a 40% (FAO, IFAD, UNICEF, WFP and WHO, 2024). Ou seja, apenas 60% da água tratada chega ao seu destino final. Além do desperdício de um recurso tão valioso, vazamentos não identificados podem levar a danos estruturais nas tubulações, aumento dos custos operacionais e insatisfação dos consumidores devido a interrupções não programadas no fornecimento de água.

No Brasil, a evolução do setor hídrico é acompanhada por relatórios anuais disponibilizados pelo Sistema Nacional de Informações sobre Saneamento (SNIS). De acordo com o relatório de 2023 (MATTOS DANIEL VALENCIA CáRDENAS, 2023), o índice de perdas na distribuição de água em 2022 foi de 37,8%, com destaque para a região Norte, onde 46,9% da água tratada é desperdiçada. Em termos quantitativos, esse índice indica que, no Brasil, a cada 100 litros de água fornecidos pelos prestadores de serviços, apenas 62,2 litros são efetivamente utilizados pelos consumidores. Esse cálculo leva em conta vazamentos, falhas no sistema de medição e ligações clandestinas. A Figura 1 representa os índices de perdas na distribuição de água de todo os estados do Brasil.

O crescimento populacional nas áreas urbanas gerou a necessidade de tornar a gestão das cidades mais inteligentes para atender às necessidades da sua população, com isso, surgiu o conceito de cidades inteligentes (*smart cities*) Talari et al. (2017), Eremia, Toma e Sanduleac (2017). Esse conceito engloba o desenvolvimento de ferramentas inteligentes

Figura 1 – Índices de perdas na distribuição de água no Brasil.



Fonte: Adaptado de Mattos Daniel Valencia Cárdenas (2023).

para coleta e análise de dados, permitindo uma abordagem mais eficiente no controle do desperdício. Dessa forma, é possível promover um crescimento sustentável e melhorar a qualidade de vida dos cidadãos por meio de uma melhor gestão dos recursos hídricos e energéticos (ARAGãO, 2020).

Neste contexto, o desenvolvimento de técnicas eficazes para a localização de vazamentos é fundamental para reduzir o desperdício e minimizar os prejuízos financeiros associados.

Assim, este trabalho propõe a aplicação de técnicas de aprendizado de máquina em estruturas de grafo, com destaque para as redes neurais para grafos (GNNs), para a detecção e localização de vazamentos em redes de distribuição de água (RDA). A proposta baseia-se na análise de dados gerados a partir de simulações realizadas na ferramenta computacional EPANET, que considera cenários com e sem a incidência de vazamentos.

O EPANET é um software desenvolvido pela Agência de Proteção Ambiental dos Estados Unidos (EPA) para simular redes de distribuição de água, permitindo a realização de análises hidráulicas e de qualidade da água (TEJASWINI; SILO; ASHWINI, 2020). Essa ferramenta é amplamente utilizada tanto por empresas do setor quanto por pesquisadores,

dada sua capacidade de simular uma variedade de cenários em sistemas de distribuição pressurizados, como falhas, expansões ou avaliações detalhadas do consumo (ELIADES et al., 2016).

A abordagem proposta aproveita a capacidade das GNNs de integrar os dados das simulações com a estrutura topológica da rede, permitindo compreender a variação dos dados em função da localização dos sensores instalados nos hidrômetros dos consumidores.

Para validar a eficácia do método proposto, modelou-se a RDA da cidade de Hanói no EPANET e com o auxílio da integração entre o EPANET e o MATLAB, realizaram-se diversas simulações que contemplaram diferentes condições de operação, considerando tanto a presença quanto a ausência de vazamentos. Os dados resultantes foram submetidos a uma etapa de pré-processamento para adequação e, em seguida, utilizados como entrada para as arquiteturas de GNN analisadas.

1.2 OBJETIVO

O objetivo deste trabalho é desenvolver uma metodologia baseada em técnicas de aprendizado de máquina em estruturas de grafo com o propósito de identificar e localizar vazamentos em redes de distribuição de água (RDAs). Para alcançar esse objetivo, o estudo analisa as características das arquiteturas G-LSTM, GGNN e GraphSAGE, compreendendo suas vantagens e limitações a partir do processamento de dados gerados por simulações realistas na ferramenta computacional EPANET, o que garante a representação de cenários reais de operação.

A validação é realizada por meio de cinco experimentos principais:

- Análise dos resultados sob condições ideais de operação;
- Análise da influência das inicializações e de falhas operacionais (falha de leitura sequencial dos sensores durante a execução);
- Análise da importância individual de cada sensor;
- Identificação dos sensores mais críticos para o treinamento dos modelos;
- Determinação do número mínimo de sensores necessários para garantir um desempenho satisfatório, incluindo a análise da influência dos melhores e piores subconjuntos de sensores selecionados.

1.3 JUSTIFICATIVA

A identificação de vazamentos em redes de distribuição de água é um tema de grande relevância, tanto do ponto de vista econômico quanto ambiental. Em muitos países,

as perdas de água não faturada devido a vazamentos representam uma fração significativa da água total distribuída, acarretando não apenas prejuízos financeiros, mas também o desperdício de um recurso natural essencial. Além disso, as mudanças climáticas e o crescimento populacional têm pressionado os sistemas de abastecimento de água, tornando a gestão eficiente desse recurso ainda mais crítica. Nesse contexto, a proposta de uma técnica que permita detectar e localizar vazamentos de forma eficaz e precisa justifica-se pela necessidade de otimizar o uso da água, reduzir perdas e garantir a sustentabilidade dos sistemas de distribuição.

A aplicação de técnicas de aprendizado de máquina, neste cenário, é uma via promissora para analisar e identificar a incidência de vazamentos em RDAs. Embora as abordagens tradicionais sejam importantes, elas esbarram em limitações ao tentar processar dados estruturados no formato de grafos. Diante disso, a utilização de modelos baseados em Redes Neurais para Grafos surge como uma solução promissora.

1.4 ESTRUTURA E CONTEÚDO

O presente trabalho está dividido nos capítulos detalhados abaixo:

A revisão bibliográfica deste trabalho está presente no **Capítulo 2** e aborda os seguintes tópicos: **a) Redes Neurais para Grafos:** explora as técnicas de aprendizado de máquina utilizadas neste trabalho para o processamento e análise de dados em forma de grafos. **b) Detecção de Vazamentos em Redes de Distribuição de Água:** oferece uma visão geral das técnicas e métodos utilizados para detectar e localizar vazamentos, desde técnicas tradicionais baseadas em modelagem hidráulica até abordagens mais recentes que utilizam aprendizado de máquina.

O **capítulo 3** apresenta os trabalhos correlatos que utilizam métodos de detecção baseados na análise de dados. Este capítulo está dividido em: **a) Trabalhos recentes de aprendizado de máquina para detecção e localização de vazamentos:** são apresentados trabalhos correlatos e estudos relevantes de outros pesquisadores na área de detecção de vazamentos em redes de distribuição de água, com ênfase nas técnicas de aprendizado de máquina e **b) Trabalhos que utilizam a RDA da cidade de Hanói:** Com intuito de analisar outros trabalhos que utilizam a RDA utilizada neste projeto este capítulo analisa diversos trabalhos que utilizam a RDA da cidade de Hanói como estudo de caso.

Os materiais e métodos desta pesquisa são abordados no **capítulo 4**, com ênfase no detalhamento do processo de elaboração das RDAs e sua simulação, utilizando a integração dos softwares EPANET e MATLAB. São descritos os procedimentos para processamento dos dados provenientes das simulações, bem como a aplicação das redes neurais desenvolvidas. O capítulo abrange desde a preparação e configuração das RDAs

até a execução das simulações, explicando os critérios adotados para o tratamento e a seleção das variáveis relevantes. Além disso, detalham-se as arquiteturas da rede neural, os parâmetros de treinamento e as métricas de avaliação utilizadas para validar o desempenho da rede. **a) Construção dos Modelos de RDAs:** Detalha o processo de construção de uma RDA utilizando o EPANET, incluindo as etapas de configuração e parametrização. **b) RDA da cidade de Hanói do Vietnã:** Detalha o dataset da cidade de Hanói descrevendo suas principais características. **c) Implementação das arquiteturas de GNN:** Detalha a implementação das GNNs utilizadas neste trabalho e **d) Métricas de Avaliação:** Explica as métricas utilizadas para medir e validar o desempenho dos modelos implementados.

O **Capítulo 5** detalha os Resultados e Discussões desta pesquisa. Este capítulo é dividido em 5 experimentos principais em que os resultados obtidos pelos modelos são analisados de acordo com métricas de desempenho para se analisar os pontos fortes e as limitações de cada um dos modelos utilizados.

No **Capítulo 6**, são apresentadas as conclusões obtidas ao longo deste trabalho. Discutem-se os resultados alcançados, as contribuições do estudo para o campo de detecção de vazamentos, bem como suas limitações e possíveis direções futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo fornecerá a base teórica necessária para o entendimento deste trabalho. Inicialmente, será realizada uma revisão detalhada das técnicas de aprendizado de máquina utilizadas neste trabalho com o intuito de abranger suas principais características. Mais adiante, serão abordadas as diferentes técnicas para detecção e localização de vazamentos de água.

2.1 REDES NEURAIS PARA GRAFOS (GNN)

Esta seção apresentará os principais conceitos das redes neurais para grafos (GNNs). A compreensão desses conceitos permitirá entender como as GNNs podem ser utilizadas para analisar padrões associados a possíveis vazamentos.

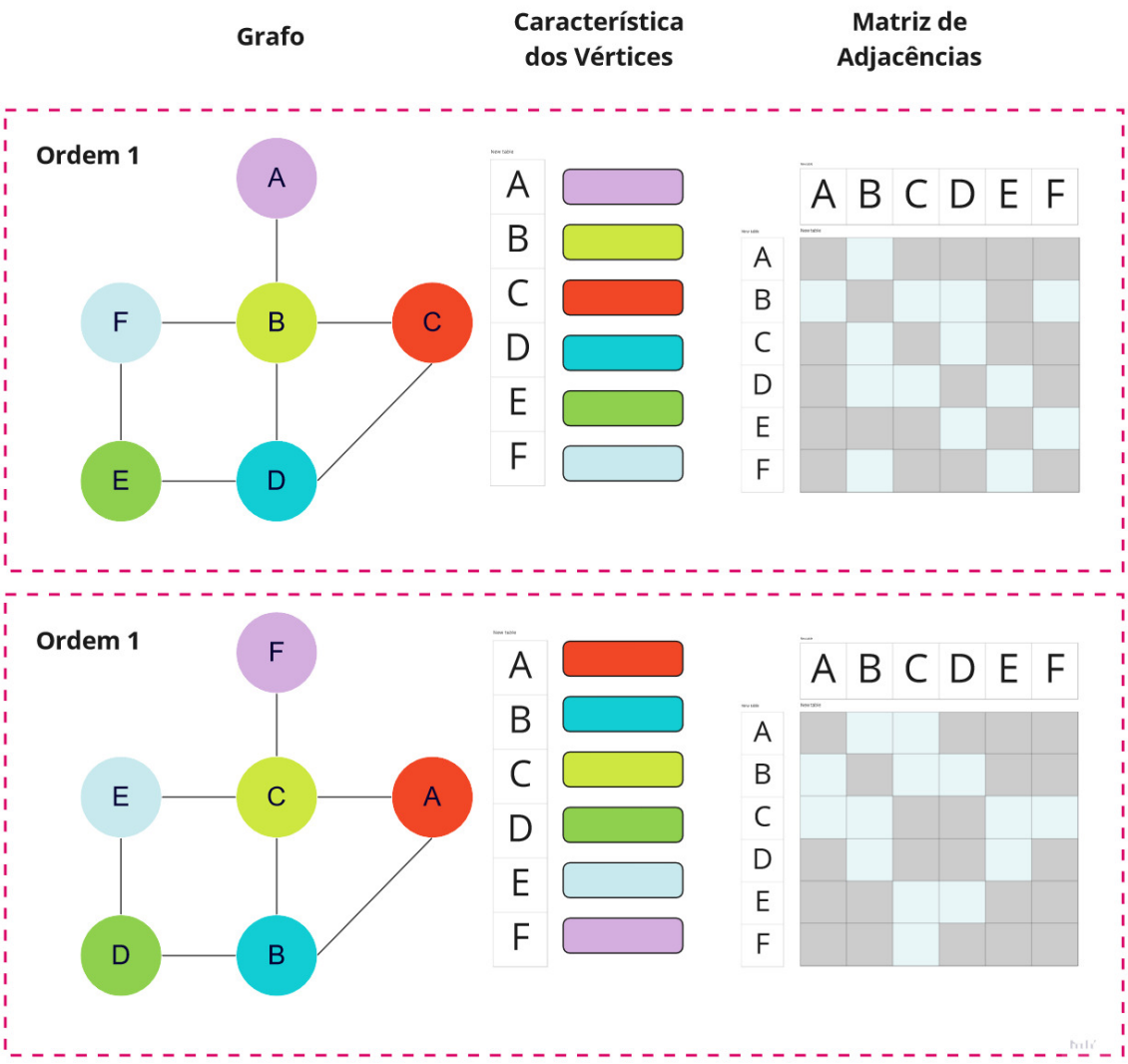
Modelos de Aprendizado Profundo (*Deep Learning*) têm demonstrado sucesso em diferentes áreas, especialmente naquelas em que os dados estão representados no espaço euclidiano. No entanto, em muitas situações do mundo real, os objetos e fenômenos de interesse são melhor descritos pela forma como se conectam e interagem entre si. Para esses casos, os grafos surgem como uma alternativa simples e intuitiva de representação, capturando tanto os objetos quanto as relações entre eles (SANCHEZ-LENGELING et al., 2021).

Entretanto, como os grafos representam dados não euclidianos e possuem informações topológicas complexas, as técnicas tradicionais de aprendizado profundo encontram dificuldades ao lidar com esse tipo de estrutura (LIU, 2020). As limitações surgem porque muitos modelos de aprendizado profundo, como redes convolucionais (CNN), foram projetados para dados que possuem uma organização rígida, como as imagens em formato de grade (*grid*). Em uma imagem, por exemplo, os píxeis (vértices) são organizados uniformemente, facilitando a análise por técnicas tradicionais, enquanto os grafos são, por natureza, estruturas de tamanho e forma arbitrários, podendo até mesmo variar no tempo e conter atributos multimodais.

Além disso, os grafos possuem uma propriedade conhecida como **invariância de permutação**, o que significa que grafos estruturalmente idênticos podem ser representados por diferentes matrizes de adjacência, dependendo da ordem dos vértices. Esse fator torna inadequado o uso de redes convolucionais tradicionais, já que um mesmo grafo pode produzir diferentes resultados dependendo da ordem de seus vértices (HAMILTON, 2020).

Para solucionar esses desafios, é comum que métodos tradicionais de aprendizado profundo utilizem técnicas de pré-processamento que mapeiem as informações topológicas

Figura 2 – Invariância da permutação em grafos.

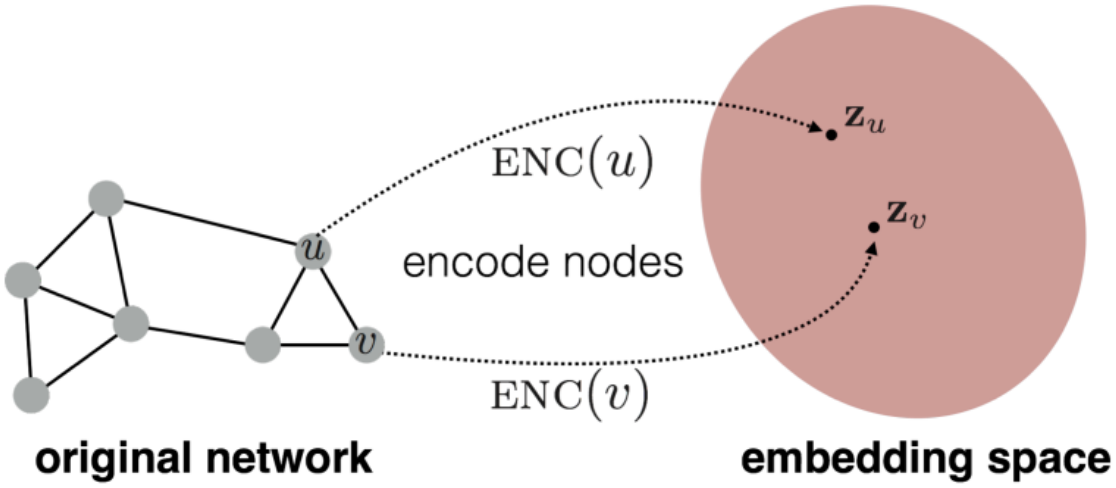


Fonte: Adaptado de Hamilton (2020).

dos grafos para uma representação mais simples. No entanto, essa abordagem pode levar à perda de informações importantes, prejudicando a eficiência e a precisão do modelo (HAMILTON; YING; LESKOVEC, 2017b; SCARSELLI et al., 2009).

As Redes Neurais para Grafos (GNNs) foram desenvolvidas para superar essas limitações, permitindo que os grafos sejam diretamente utilizados como entrada sem necessidade de tratamento extra (HAMILTON; YING; LESKOVEC, 2017b). Esses modelos transformam os grafos em um espaço n-dimensional, no qual os vértices com características similares são mapeados para ficarem próximos um do outro, em um processo conhecido como *embedding* (XU et al., 2019). Isso significa que os vértices próximos no espaço de *embedding* apresentam um alto grau de similaridade. A Figura 3 demonstra o funcionamento do processo de *embedding*.

Figura 3 – Processo de *embedding* das GNN.



Fonte: Adaptado de Hamilton (2020).

O *encoder* representado na Figura 3 pode ser qualquer arquitetura de GNN, e seu objetivo é mapear os vértices para um novo espaço (*embedding space*). Nessa etapa, é definida uma função de similaridade que considera aspectos como distância e características dos vértices para identificar as similaridades entre eles.

Algumas arquiteturas utilizam uma estratégia simples, onde as informações são agregadas com base na vizinhança de cada vértice em um processo iterativo. Esse processo atualiza a representação dos vértices com base nas informações contidas em seus vizinhos, e termina quando todos os vértices foram processados ou quando um ponto de parada específico é atingido (XU et al., 2019).

Este processo de agregação de informações a partir dos vizinhos é denominado ***message passing***. Nesse mecanismo, cada vértice do grafo é tratado como uma unidade conectada conforme as respectivas arestas, atualizando seu estado e trocando informações até que se alcance um equilíbrio. A saída final é computada localmente em cada vértice, com base no estado alcançado pela unidade (SCARSELLI et al., 2009). Ou seja, os vértices possuem representações vetoriais as quais são utilizadas para trocar informações entre si. Desta forma, cada vértice, além de suas próprias informações, também possui informações sobre as características de seus vizinhos.

Seja um grafo indireto $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ onde cada vértice $\mathbf{v_i}$ possui um vetor de atributos $\mathbf{x_i}$ e cada aresta $\mathbf{v_i}, \mathbf{v_j}$ é caracterizada por um vetor de atributos $\mathbf{e_{ij}}$. A atualização dos estados ocorre por camadas $\mathbf{k}$, assim, o estado oculto do vértice $\mathbf{v_i}$ na camada $\mathbf{k}$ é denotado pela representação vetorial $\mathbf{h_i^{(k)}}$.

A cada camada, realiza-se um processo de atualização e agregação para combinar informações da vizinhança de $\mathbf{v_i}$, conforme definido na Equação 2.1. Onde, $\mathbf{c_i^{(k)}}$ representa

a atualização das informações da vizinhança de $\mathbf{v_i}$ e é combinado com o vetor de atributos $\mathbf{x_i}$ do nó para produzir uma nova representação $\mathbf{h_i^{(k+1)}}$ para a próxima camada.

$$c_i^{(k)} = \text{AGREGAR}^{(k)}\Big(h_j^{(k)} : j \in N(i)\Big). \tag{2.1}$$

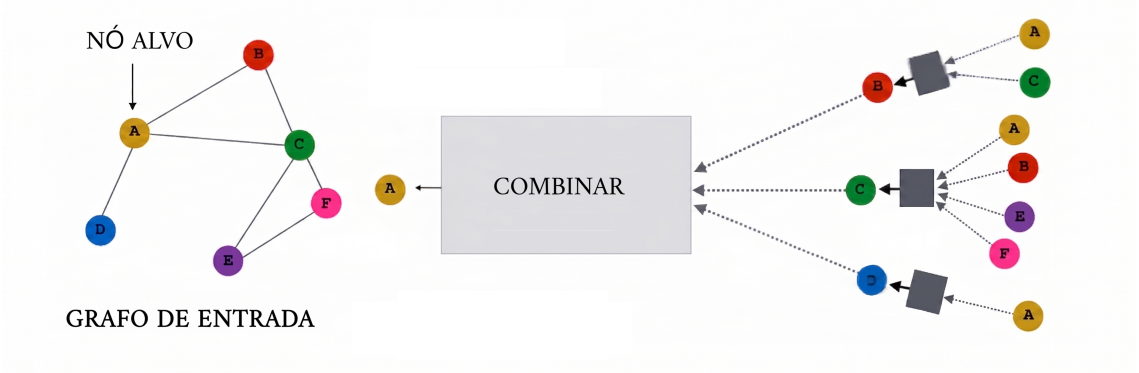
Nesta Equação, $\mathcal{N}(\mathbf{i})$ representa o conjunto de vizinhos do nó $\mathbf{v_i}$ e $\text{AGREGAR}^{(\mathbf{k})}$ é uma função de atualização, como soma, média, ou máximo, utilizada para combinar as representações dos vizinhos. Após a etapa de atualização, a nova representação do nó $\mathbf{v_i}$ para a camada $(\mathbf{k+1})$ é calculada por meio da função $\text{COMBINAR}^{(\mathbf{k})}$, que combina $\mathbf{c_i^{(k)}}$ com $\mathbf{h_i^{(k)}}$ e $\mathbf{x_i}$, conforme

$$h_i^{(k+1)} = \text{COMBINAR}^{(k)}\Big(h_i^{(k)}, c_i^{(k)}, x_i\Big). \tag{2.2}$$

$\text{COMBINAR}^{(\mathbf{k})}$ pode ser qualquer função de combinação, soma, etc.

A Figura 4 ilustra o processo de atualização de mensagens de um único nó a partir de sua vizinhança local. O modelo agrega mensagens dos vizinhos diretos do nó A (ou seja, os nós B, C e D). Por sua vez, as mensagens provenientes destes vizinhos baseiam-se em informações previamente agregadas de suas próprias vizinhanças, e assim sucessivamente.

Figura 4 – Mecanismo de *message passing*.



Fonte: Adaptado de Hamilton (2020).

Nos últimos anos, as redes neurais para grafos vêm ganhando popularidade devido aos seus resultados promissores em diversas áreas de aplicação. Diante disso, múltiplas arquiteturas foram desenvolvidas com o intuito de resolver problemas específicos e aperfeiçoar determinadas características operacionais. Na sequência, serão detalhadas as arquiteturas adotadas neste trabalho.

2.1.1 *LONG SHORT-TERM MEMORY GRAPH NEURAL NETWORK (G-LSTM)*

As Redes Neurais Recorrentes (RNNs) são uma classe de modelos de aprendizado de máquina projetada para lidar com dados sequenciais. A principal característica que as distingue das redes neurais *feedforward* tradicionais é a presença de conexões recorrentes em sua arquitetura, que geralmente é composta por uma camada de entrada, uma camada oculta e uma camada de saída. Essas conexões recorrentes permitem que a informação circule continuamente dentro da rede, conferindo às RNNs a capacidade única de manter uma ”memória” das entradas anteriores.

O treinamento dessas redes é realizado por meio do algoritmo de retropropagação no tempo (*Backpropagation Through Time* - BPTT) (MIENYE; SWART; OBAIDO, 2024). No entanto, ao processar sequências muito longas, essas redes sofrem com uma limitação fundamental conhecida como o problema do desaparecimento do gradiente (*vanishing gradient*). Durante o processo de BPTT, o cálculo do gradiente da função de perda em relação aos pesos envolve multiplicações sucessivas das matrizes de pesos e das derivadas das funções de ativação ao longo das etapas de tempo. Se essas derivadas forem menores que 1 o gradiente decai exponencialmente, tornando-se irrelevante nas etapas iniciais. Isso impede que a rede atualize os parâmetros de forma satisfatória (BENGIO; SIMARD; FRASCONI, 1994).

Para mitigar essa limitação, a LSTM utiliza um mecanismo de portas (*gates*) para o controle do fluxo de informações e a manutenção de um estado de célula (*cell state*) que garante o fluxo ininterrupto do gradiente durante a retropropagação. A atualização desse estado é controlada por três mecanismos de portas (*gates*): a porta de esquecimento (*forget gate*), que descarta informações obsoletas; a porta de entrada (*input gate*), que incorpora novos dados relevantes; e a porta de saída (*output gate*), que filtra a informação repassada ao próximo estado oculto (*hidden state*) da rede (HOCHREITER; SCHMIDHUBER, 1997).

Esse mecânismo pode ser adaptado para atuar diretamente no processo de propagação de mensagens em grafos, regulando como cada vértice assimila as informações de sua vizinhança.

Na etapa de inicialização o portão de esquecimento (*forget gate*) f_v^t controla a quantidade de informações da etapa anterior qual deve ser descartada, conforme Equação 2.3

$$f_v^t = \sigma(W_f a_v^t + U_f h_v^{(t-1)} + b_f) \tag{2.3}$$

O portão de entrada (*input gate*), i_v^t, determina quanta informação nova deve ser

armazenada no estado da célula, enquanto $\tilde{c}_v^t$ representa os valores candidatos para a atualização da memória, demonstrado nas equações 2.4 e 2.5 respectivamente

$$i_v^t = \sigma(W_i a_v^t + U_i h_v^{(t-1)} + b_i) \tag{2.4}$$

$$\tilde{c}_v^t = \tanh(W_c a_v^t + U_c h_v^{(t-1)} + b_c) \tag{2.5}$$

Com os resultados de f_v^t e i_v^t, o novo estado da memória da célula (c_v^t) é calculado:

$$c_v^t = f_v^t \odot c_v^{(t-1)} + i_v^t \odot \tilde{c}_v^t \tag{2.6}$$

O termo o_v^t atua como o portão de saída (*output gate*) e controla qual proporção do estado da célula deve ser repassada ao estado oculto e em seguida, o estado oculto h_v^t é atualizado, conforme equações 2.7 e 2.8.

$$o_v^t = \sigma(W_o a_v^t + U_o h_v^{(t-1)} + b_o) \tag{2.7}$$

$$h_v^t = o_v^t \odot \tanh(c_v^t) \tag{2.8}$$

2.1.2 *GATED GRAPH NEURAL NETWORK (GGNN)*

As primeiras abordagens de redes neurais recorrentes aplicadas a grafos, como apresentado por Gori, Monfardini e Scarselli (2005), baseavam-se em uma representação de ponto fixo e aprendizado via mapas de contração. Entretanto, Li et al. (2015) apontam que essas abordagens limitam a capacidade de aprendizado, dificultando a captura de relações de longa distância entre vértices.

Para superar essa limitação, Li et al. (2015) propuseram a *Gated Graph Neural Network* (GGNN), que combina a estrutura da rede proposta por Gori, Monfardini e Scarselli (2005) e Scarselli et al. (2009) com mecanismos de *gating* presentes nas *Gated Recurrent Units (GRUs)*. Esse mecanismo permite o aprendizado por meio de um processo de *backpropagation through time (BPTT)*. Esse método exige mais memória, mas elimina a necessidade de restringir parâmetros para garantir convergência (GORI; MONFARDINI; SCARSELLI, 2005).

A propagação na GGNN ocorre de forma recorrente, conforme ilustrado nas Equações 2.9 a 2.14.

$$h_v^1 = [x_v^T, 0]^T, \tag{2.9}$$

onde h_v^t é o estado oculto do vértice v no tempo t, e x_v é o vetor de anotações do vértice v.

$$a_v^t = A_v^T[h_1^{(t-1)T}, \ldots, h_{|V|}^{(t-1)T}]^T + b \tag{2.10}$$

Cada vértice v coleta informações de seus vizinhos. a_v^t é o vetor agregado de informações dos vizinhos de v no tempo t, e A_v é uma matriz derivada da matriz de adjacência que indica as conexões de v. V é o número total de vértices no grafo

$$z_v^t = \sigma\left(W^z a_v^t + U^z h_v^{(t-1)}\right) \tag{2.11}$$

$$r_v^t = \sigma\left(W^r a_v^t + U^r h_v^{(t-1)}\right) \tag{2.12}$$

Nas Equações 2.11 e 2.12, z e r representam os portões de atualização e de *reset*, respectivamente, em uma estrutura semelhante às GRUs. Os parâmetros W^z, U^z, W^r e U^r são usados para computar esses portões, enquanto σ representa a função sigmoide.

$$\widetilde{h_v^t} = \tanh\left(W a_v^t + U\left(r_v^t \odot h_v^{(t-1)}\right)\right) \tag{2.13}$$

$$h_v^t = (1 - z_v^t) \odot h_v^{(t-1)} + z_v^t \odot \widetilde{h_v^t} \tag{2.14}$$

Onde $\tanh(x)$ é a função de ativação tangente hiperbólica, e $\odot$ representa uma multiplicação elemento por elemento.

Esse processo de atualização usa o estado anterior de cada vértice e a informação agregada dos vizinhos para atualizar o estado oculto do vértice.

As demais equações (2.11, 2.12, 2.13 e 2.14) são responsáveis pela atualização dos estados ocultos por meio da unidade GRU, incorporando tanto as informações dos vértices vizinhos quanto do tempo anterior, para realizar a atualização do estado oculto de cada vértice (LI et al., 2015).

Resumidamente, a GGNN inicializa os estados ocultos h_v de cada vértice para corresponder à sua anotação x_v. Em seguida, ocorrem as etapas de propagação, nas quais as informações são transferidas entre os vértices por meio das arestas do grafo. Cada vértice soma os valores recebidos de seus vizinhos para atualizar seu próprio estado interno, de maneira similar ao processo realizado em uma GRU tradicional. Finalmente, os estados de todos os vértices são utilizados para gerar uma saída, que pode ser usada tanto para classificar o grafo inteiro quanto para realizar classificações individualizadas para cada vértice (JOHNSON, 2016).

Em problemas de grafos que exigem uma sequência de predições, Li et al. (2015) propuseram a *Gated Graph Sequence Neural Network* (GGSNN), uma extensão da GGNN que realiza predições sequenciais. Cada etapa de predição produz uma saída e uma nova anotação para os vértices, que servem como entrada para a próxima etapa. (HAMILTON, 2020)

2.1.3 *GRAPHSAGE*

A maioria das arquiteturas tradicionais de GNN requer o acesso a toda a estrutura do grafo para gerar representações. Para superar essa limitação, Hamilton, Ying e Leskovec (2017a) propuseram o GraphSAGE (Graph Sample and Aggregate), uma arquitetura de natureza indutiva, capaz de generalizar o aprendizado para nós não vistos durante o treinamento. Por meio de um processo de amostragem e agregação, o modelo gera representações vetoriais (*embeddings*) de forma eficiente, adequando-se a grafos extensos e dinâmicos.

A fim de manter a eficiência computacional, o GraphSAGE não atualiza as representações utilizando a vizinhança completa de um nó. Em vez disso, o método extrai uma amostra de tamanho fixo dos vizinhos de cada vértice. As informações *(features)* desses vizinhos amostrados são então agregadas e combinadas para atualizar a representação do vértice alvo. Essa estratégia de amostragem garante uma complexidade computacional constante e viabiliza a aplicação do algoritmo em grafos de grande escala.

A etapa de agregação é descrita pelas Equações 2.15 e 2.16:

$$h^k_{N_v} = \text{AGGREGATE}_k \left(h^{k-1} u, \forall u \in N_v \right) \tag{2.15}$$

$$h^k_v = \sigma \left(W^k \cdot \left[h^{k-1} v \parallel h^k N_v \right] \right) \tag{2.16}$$

O processo em cada etapa procede da seguinte forma, onde k representa a etapa atual e h^k a representação de um nó nesta etapa. De início, cada $v \in V$ agrega as representações da sua vizinhança imediata amostrada $\left(\{ h^{k-1}_u, \forall u \in N_v \} \right)$ em um único vetor $h^k_{N_v}$. Após agregar os vetores de características da vizinhança, o GraphSAGE concatena a representação atual do próprio nó h^{k-1}, com o vetor agregado da vizinhança, $h^k_{N_v}$. Esse vetor concatenado é então multiplicado por uma matriz de pesos W^k e submetido a uma função de ativação não linear σ, resultando na representação atualizada h^k_v

As funções de agregação podem apresentar diferentes tipos de arquiteturas. O artigo original deste método utiliza três funções diferentes: *pooling*, LSTM e média (HAMILTON; YING; LESKOVEC, 2017a). Neste trabalho serão aplicadas as mesmas variações, conforme descrito a seguir:

- **Média**: A função de agregação da média é simillar à regra de propagação convolucional utilizada na arquitetura da rede neural convolucional em grafos (*Graph Convolutional Networks - GCN)* transdutiva (KIPF; WELLING, 2017).

$$h_{N_v}^k = \sigma\left(W \cdot \text{MÉDIA}\left(\{h_v^{k-1}\} \cup \{h_u^{k-1}, \forall u \in N_v\}\right)\right) \tag{2.17}$$

  Essa função é mais simples que as outras funções utilizadas para agregação pois não realiza a concatenação entre h_v^{k-1} e $h_{N_v}^k$. Este processo pode ser interpretado como uma conexão de salto *skip connection* (RODRIGUES; ROLLE; GODOY, 2024).

  Este tipo de conexão cria um caminho direto para a propagação da informação por toda a extensão do modelo. O mecanismo central dessa técnica baseia-se em um mapeamento de identidade, no qual o vetor de entrada original é somado diretamente à saída de uma função de transformação. Esta estrutura impede a degradação do sinal, permitindo que a informação flua de maneira contínua e direta (HE et al., 2016).

- **LSTM**: O *Long Short-Term Memory* (LSTM) foi empregado devido à sua alta capacidade de processamento. No entanto, para ser utilizado como uma função de agregação, foi necessária uma adaptação. Para que o algoritmo consiga operar em um conjunto não ordenado de nós, ao invés de processar entradas estritamente sequenciais como na arquitetura original, a rede realiza uma permutação dos nós vizinhos (HAMILTON; YING; LESKOVEC, 2017a).

- ***Pooling***: a função de agregação *pooling* aplica uma rede neural simples para processar cada vizinho e, em seguida, utiliza a operação de *max-pooling* para capturar as informações mais importantes de toda a vizinhança. Esse processo simplifica as informações oriundas da vizinhança, aprimorando o aprendizado do modelo sem a perda de características relevantes.

$$h_{N_v}^k = \max\left(\left\{\sigma\left(W_{\text{pool}} h_u^{k-1} + b\right), \forall u \in N_v\right\}\right)$$

Ressalta-se que a operação de *max-pooling* pode ser substituída por qualquer função simétrica capaz de processar um número variável de entradas e produzir um vetor de tamanho fixo, a exemplo da média, da soma e do mínimo.

2.2 DETECÇÃO DE VAZAMENTOS EM RDAs

As RDAs são sistemas interconectados que podem ser perfeitamente modelados como grafos, nos quais os nós representam os seus componentes físicos (bombas, reserva-

tórios, junções e consumidores) e as arestas correspondem às tubulações que interligam esses equipamentos.

Compreender essa estrutura é fundamental, pois a detecção de vazamentos em RDAs é uma tarefa complexa, visto que a maioria das falhas são ocultas e não visíveis (COMMISSION; DIRECTIVE, 2020). A presente seção irá demonstrar as principais características dos métodos de detecção e localização de vazamentos.

No decorrer da evolução humana, muitas cidades se desenvolveram próximas às margens de rios e lagos devido à dependência humana pela água, seja para uso pessoal ou para processos industriais que se desenvolveram ao longo dos anos.

Com o passar do tempo e o crescimento das cidades, aumentou-se a distância entre as fontes de água e os consumidores finais. Para suprir essa necessidade, foram desenvolvidas as primeiras RDAs, entre 3200 e 1100 a.C. (ANGELAKIS LARRY W. MAYS, 2012). Isso inclui um sistema composto por centenas de poços na Grécia. Posteriormente, entre os séculos VIII e VI a.C., os gregos construíram uma série de túneis e pontes exclusivamente para o transporte de água (CROUCH, 1993). Entre 100 a.C. e 500 d.C., os romanos desenvolveram um sistema de captação de água das fontes e a distribuíam através de tubulações de madeira ao longo da cidade de Roma, alcançando distâncias de mais de 100 km (MAYS; KOUTSOYIANNIS; ANGELAKIS, 2007). Diversas civilizações, ao longo do tempo, criaram métodos para transportar água e suprir suas necessidades.

Figura 5 – Aqueduto Pont du Gard, na França, patrimônio mundial da UNESCO.



Fonte: National Geographic Society (2025).

Em 1664, na França, foi construída uma linha de tubos interligando Marle-on-Seine ao Palácio de Versalhes, considerada por muitos como a primeira grande rede de

distribuição de água (WALSKI, 2006). Com o passar dos anos, os métodos de distribuição de água evoluíram: a utilização de canos de madeira foi substituída por materiais como ferro, PVC e concreto, tornando o transporte de água cada vez mais eficiente. Contudo, essa eficiência é altamente afetada por diversos problemas, entre eles, os vazamentos, que são os principais responsáveis pelas perdas nas RDAs.

2.2.1 REDES DE DISTRIBUIÇÃO DE ÁGUA (RDAs)

São responsáveis por transportar água potável, proveniente de algum reservatório, até o consumidor final, assim, são essenciais para garantir o fornecimento de água em áreas de pequeno, médio e grande porte. Estas redes são compostas por uma rede de tubulações e outros componentes tais como reservatórios, bombas, etc.

2.2.1.1 PERDAS NAS RDAs

De acordo com o Banco Mundial, 45 bilhões de metros cúbicos de água são desperdiçados diariamente nos países em desenvolvimento, o que representa aproximadamente 3 bilhões de dólares em perdas financeiras (TORRES et al., 2020).

No trabalho de Hirner e Lambert (2000), as perdas são classificadas em duas categorias distintas:

1. consumo não autorizado (fraudes) ou erros de medição;
2. perdas físicas decorrentes de vazamentos ou rompimento de tubulações.

As perdas físicas podem ter diversas causas, como desgaste natural devido à corrosão, deterioração causada pela falta de manutenção adequada, variações bruscas na pressão da rede, fadiga, fissuras, entre outros (ROJEK; STUDZINSKI, 2019).

Os rompimentos de tubulações e grandes vazamentos são, geralmente, mais fáceis de detectar, uma vez que costumam causar problemas visíveis, como deslocamento de solo e áreas de umidade excessiva, além de variações perceptíveis em variáveis de controle, como volume e vazão. Em contraste, pequenos vazamentos apresentam maior dificuldade de detecção, pois, além de demorarem mais para gerar sinais visíveis de dano, quase não afetam as variáveis de controle de forma significativa, prolongando sua identificação e permitindo que permaneçam despercebidos por mais tempo.

Além dos desafios de detecção, é importante destacar que os vazamentos representam cerca de 80% das perdas de água nas redes de distribuição, o que corresponde a um valor de 500 a 1000 vezes o gasto pelos consumidores domésticos (ROGERS, 2014). A fim de reduzir os desperdícios resultantes de vazamentos, é essencial monitorar constantemente

as redes de distribuição de água. Ao longo dos anos, diversas técnicas foram desenvolvidas para diminuir o tempo de identificação.

2.2.2 MÉTODOS PARA DETECÇÃO DE VAZAMENTOS EM RDAs

Como mencionado anteriormente, os vazamentos representam a principal causa de desperdício de água nas RDAs. Para mitigar esse problema, diversas técnicas foram desenvolvidas com o objetivo de analisar vazamentos. Contudo, apesar dos inúmeros estudos, ainda não há uma técnica universalmente consagrada; ao contrário, existem múltiplas abordagens cujo desempenho varia conforme as condições específicas de cada situação.

Diversos pesquisadores têm proposto diferentes formas de categorizar essas técnicas. Scott e Barrufet (2003) por exemplo, sugeriram uma divisão em dois subgrupos: **a) métodos externos** que requerem a instalação de sensores adicionais para monitorar a presença de vazamentos, e **b) métodos internos** que utilizam sensores já presentes na rede (como medidores de vazão e pressão) para identificar vazamentos. Embora essa categorização seja de fácil entendimento, ela não é amplamente utilizada em trabalhos mais recentes.

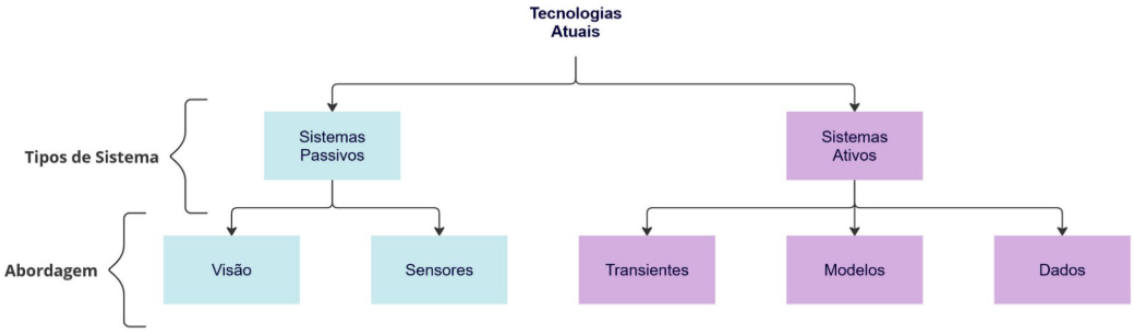
Já Puust et al. (2010) propuseram uma categorização em duas classes principais: **a) Modelagem Hidráulica** que consiste na criação de um modelo simplificado dos elementos da rede (como tubulações, bombas e reservatórios) e utiliza simulações do comportamento normal e anormal (com vazamentos) do sistema para identificar falhas, e **b) Análise de Sinais** que busca detectar vazamentos através da alteração de variáveis específicas do sistema, baseando-se em padrões de comportamento. Os sinais podem ser captados tanto durante a operação regular do sistema quanto em condições transitórias.

Em uma abordagem mais recente, Torres et al. (2020), Chan, Chin e Zhong (2018) propuseram uma nova divisão, agrupando os métodos em **passivos** e **ativos**. Os **métodos passivos** dependem de **inspeção visual** ou monitoramento direto do sistema, enquanto os **métodos ativos** utilizam a **análise de sinais**, como dados sonoros ou de pressão. A Figura 6 ilustra essas abordagens, destacando as principais técnicas de cada categoria.

Os métodos de detecção passiva de vazamentos podem ser classificados em duas categorias principais. A primeira envolve a **inspeção visual** direta do local, onde se busca por sinais evidentes que indicam a presença de vazamentos, como o crescimento anormal de vegetação, degradação do solo, formação de poças d'água, entre outros. A segunda categoria envolve o uso de **sensores portáteis**, operados por profissionais especializados, para detectar anomalias nas medições que possam apontar vazamentos.

Embora esses **métodos passivos** possam apresentar bons resultados na localização de vazamentos, há limitações claras. Um dos principais desafios é que os **vazamentos**

Figura 6 – Demonstração dos métodos passivos e ativos para identificação de vazamentos em RDAs.



Fonte: Adaptado de Chan, Chin e Zhong (2018).

precisam estar ativos no momento da inspeção, o que inviabiliza o monitoramento contínuo do sistema (CHAN; CHIN; ZHONG, 2018).

Os métodos de detecções ativas não necessitam inspeção direta do sistema, conforme demonstrado na Figura 6 são divididos em 3 categorias principais:

1. Métodos **baseados em transientes**, estes métodos buscam identificar o vazamento através do **transiente hidráulico**. O transiente hidráulico é uma onda de pressão de curta duração que ocorre no momento da ocorrência do vazamento (COLOMBO; LEE; KARNEY, 2009).

2. Métodos **baseados em modelos hidráulicos** buscam modelar o sistema por meio de funções matemáticas. Apesar de alcançar bons resultados, é necessário que as equações representem o sistema com um elevado grau de precisão, o que torna este modelo complexo quando aplicado em redes de distribuição grandes ou com diversos componentes (CHAN; CHIN; ZHONG, 2018).

3. Métodos **baseados em dados** (*Data-driven*): esses métodos dependem da coleta de dados por sensores distribuídos na RDA, sem a necessidade de um alto nível de detalhamento exigido pelos métodos baseados em modelos. Diferente de muitas outras abordagens, os métodos baseados em dados não requerem um profissional com profundo conhecimento sobre o comportamento hidráulico da rede. Normalmente, realizam-se análises estatísticas ou aplicam-se métodos de aprendizado de máquina para reconhecimento de padrões e monitoramento da incidência e localização dos vazamentos (CHAN; CHIN; ZHONG, 2018).

Algumas tecnologias recentes utilizam técnicas de IoT (*Internet of Things*) para transmitir de maneira *online* os valores coletados por sensores espalhados em diversos

pontos da rede. Essas tecnologias minimizam a ocorrência de interrupções de serviço e facilitam a determinação do local do vazamento (LOUREIRO et al., 2015).

3 TRABALHOS CORRELATOS

Neste capítulo, serão apresentados trabalhos recentes que exploram diferentes abordagens e algoritmos para a detecção e localização de vazamentos. A pesquisa foi realizada utilizando majoritariamente a base de dados do Google Scholar.

A estratégia principal de pesquisa concentrou-se no termo "*water distribution network leak detection*". Para garantir uma maior abrangência da busca, utilizou-se o cruzamento alternado de palavras-chave complementares como "WDN" (*Water Distribution Network*), "WDS" (*Water Distribution System*), "*Leakage*" e "*locate*". Adicionalmente, por se tratar de um dos principais *benchmarks* da área e da rede de distribuição de água utilizada neste trabalho, incorporou-se a palavra-chave "Hanoi" às strings de busca, permitindo mapear especificamente os estudos e algoritmos aplicados à RDA da cidade de Hanói.

Com o uso de sensores sem fio e técnicas de aprendizado supervisionado, Rashid, Akram e Khan (2015) desenvolveram um método para identificar o local e estimar o tamanho de vazamentos, baseado na análise da **onda de pressão negativa (NPW)**. Os experimentos foram realizados em um sistema simulado de distribuição de água, onde válvulas de diferentes dimensões simularam vazamentos com diâmetros variando de 0,25 a 1,5 polegadas. Os dados de pressão coletados foram utilizados para treinar vários algoritmos, incluindo KNN, GMM, SVM e Naive Bayes. O algoritmo SVM destacou-se, apresentando uma média de detecção de 94,5%.

Liu et al. (2019) propuseram um sistema de monitoramento destinado à detecção de vazamentos, que faz uso de sensores sem fio combinados com a técnica de máquina de Vetores de Suporte (SVM). Nesse sistema, os sensores são estrategicamente distribuídos ao longo da rede de distribuição. Antes do início do treinamento, os dados coletados passam por uma fase de pré-processamento, que visa a extração de características relevantes. Esses dados pré-processados são, então, utilizados como entrada para o classificador SVM. Os resultados demonstraram que este método alcançou uma impressionante acurácia de 98% na classificação dos dados em duas categorias: vazamentos e não vazamentos.

Utilizando simulações de redes de distribuição de água na ferramenta computacional EPANET, Mashhadi et al. (2021) propuseram um método baseado no monitoramento de vazão e pressão. A rede de distribuição modelada representa o campus da Universidade de Lille na França, dividido em cinco zonas territoriais. Diversas simulações de vazamentos foram realizadas em cada uma dessas zonas, com o objetivo de prever a zona em que o vazamento ocorre. Foram aplicados métodos supervisionados e não supervisionados. Nos métodos supervisionados, tanto a regressão logística quanto o Random Forest obtiveram

uma acurácia de 100%. Além disso, uma rede neural artificial também alcançou uma acurácia de 100%, demonstrando a eficácia da abordagem utilizada.

Fan e Yu (2022) desenvolveram um método que integra a localização topológica da rede de distribuição de água com suas características de vazamentos, visando a partição da rede e a instalação de sensores. Os dados monitorados são utilizados para prever e localizar vazamentos. Este método obteve uma acurácia de 95% na detecção de vazamentos e 85% na localização. Resumidamente, o método aplica o algoritmo k-means para definir os centroides das zonas da rede (sub-redes). A detecção de vazamentos é realizada por meio de duas técnicas distintas: PCA e uma rede neural do tipo autoencoder. Por fim, a localização do vazamento é determinada por várias técnicas, sendo que a partição baseada no PCA gerou os melhores resultados.

Fan, Zhang e Yu (2021) propuseram um modelo de aprendizado de máquina para detecção de vazamentos focado em contornar o problema prático de dados desbalanceados, já que as redes de distribuição operam sob condições normais na maior parte do tempo e possuem poucos registros de falhas. Para resolver isso, os autores desenvolveram um método não supervisionado baseado em uma rede neural do tipo Autoencoder. O modelo foi treinado exclusivamente com dados de pressão da rede em funcionamento normal, possuindo o objetivo matemático de comprimir e depois reconstruir esses mesmos dados na sua saída. A identificação de vazamentos ocorre justamente pela análise dos resíduos, ou seja, o erro de reconstrução (calculado pelo Erro Quadrático Médio - MSE) gerado pela diferença entre a pressão original inserida no modelo e a pressão recriada por ele. Quando ocorre um vazamento, a alteração no fluxo e na pressão faz com que os dados fujam do padrão que a rede aprendeu. Consequentemente, o modelo falha em reconstruir essa informação, gerando um resíduo elevado. Ao ultrapassar um limiar de tolerância (threshold) pré-definido, esse erro residual alto dispara o alerta de vazamento. Para refinar ainda mais o método e mitigar a incidência de alarmes falsos, os autores implementaram uma estratégia baseada em múltiplas tentativas de detecção independentes (um sistema de votação), o que elevou a acurácia do modelo para níveis próximos a 100%.

A utilização de técnicas de GNN é explorada em alguns artigos, Rolle et al. (2022a) foca na aplicação de machine learning em cenários de cidades inteligentes (smart cities), onde os dispositivos utilizam o protocolo LPWAN. O estudo utiliza um Perceptron de Múltiplas Camadas (MLP) para classificar dados de pressão coletados em intervalos de 10 minutos e, quando aplicado a uma RDA fictícia, a metodologia alcançou uma acurácia de 91,55% na identificação da localização dos vazamentos.

Os valores de pressão são monitorados no estudo de Rolle et al. (2022b) e na ocorrência de uma falta, a GGNN indica a probabilidade de cada nó ser o local da falta, sendo que a indicação pode ser de um único nó ou possíveis nós (normalmente na mesma região), quando não é possível identificar o nó exato. Este estudo alcançou uma acurácia de

78%. A integração entre aprendizado de máquinas e LPWAN também é explorada em Rolle et al. (2024), entretanto o diferencial desta pesquisa é na integração de regressão linear para prever o valor da pressão nos nós de junção contornando a necessidade intrínseca da maioria das técnicas de GNN de ter medições de todos os nós da rede.

A técnica GraphSAGE foi utilizada por Rodrigues, Rolle e Godoy (2024) para localizar vazamentos em RDAs, os autores utilizaram três diferentes agregadores (média, pooling e LSTM) com destaque para a utilização do LSTM como agregador que apresentou a melhor desempenho geral atingindo uma acurácia de 83,32% na validação, superando arquiteturas anteriores como a GGNN (78,50%).

Em sua tese de doutorado, Rolle (2024) desenvolveu uma metodologia completa para o monitoramento distribuído de Redes de Distribuição de Água (RDA) baseada no paradigma da Internet das Coisas (IoT). O trabalho aborda o desafio de detectar e localizar vazamentos sob restrições severas de largura de banda, típicas de redes LPWAN. A pesquisa utilizou GGNN para detecção e localização de vazamentos e também fez o uso de regressão linear para estimar os valores nos nós de junção.

Em suma, devido às dificuldades enfrentadas para se obter dados reais de vazamentos, as simulações hidráulicas computacionais com técnicas de Aprendizado de Máquina têm se consolidado como uma abordagem eficaz para a detecção e localização de vazamentos. Destaca-se o trabalho de Fan, Zhang e Yu (2021), em que o modelo foi treinado apenas com dados de simulações sem vazamentos, utilizando uma rede neural do tipo autoencoder para analisar a diferença residual entre os valores de operação normal e os de anomalia. Nesta dissertação, uma abordagem baseada em resíduos é utilizada na etapa de pré-processamento. Contudo, em vez de focar em um resíduo gerado pelo modelo, a análise de resíduos proposta é aplicada aos próprios dados de medição. Especificamente, o nó de entrada da RDA é adotado como um nó de referência, e os valores dos demais nós são computados como a diferença (ou resíduo) de suas medições originais em relação a este nó principal.

Os estudos de Rolle et al. (2022a), Rolle et al. (2022b), Rolle et al. (2024), Rodrigues, Rolle e Godoy (2024) serviram como inspiração para esta pesquisa. Além das arquiteturas exploradas anteriormente, como GraphSAGE e GGNN, o presente trabalho propõe a utilização de uma Rede Neural de Grafo (GNN) que integra unidades de memória de longo e curto prazo (LSTM). Adicionalmente, esta investigação aprofunda o entendimento sobre o comportamento dessas arquiteturas sob variadas condições de operação.

3.1 TRABALHOS QUE UTILIZAM A RDA DA CIDADE DE HANÓI

A rede de distribuição de água (RDA) da cidade de Hanói, introduzida por Fujiwara e Khang (1990), é amplamente reconhecida como um modelo de referência no estudo de detecção e localização de vazamentos em sistemas urbanos complexos. Sua estrutura oferece um cenário ideal para validação de métodos, permitindo a simulação de condições realistas, como variações de demanda, ruídos em medições de pressão e diferentes magnitudes de vazamento. Nesta seção, destacam-se trabalhos que utilizam essa rede como *benchmark* para avaliar o desempenho das técnicas desenvolvidas.

O artigo Ponce, Castañón e Cayuela (2013) analisou vazamentos em redes de água utilizando cinco métodos: binarização da matriz de sensibilidade Pérez et al. (2011), correlação dos vetores de pressão, ângulo entre vetores, distância euclidiana e otimização por mínimos quadrados. As amostras foram coletadas a cada 15 minutos, com vazamentos de 20 a 80 L/s e ruídos de ±2% a ±4% na demanda, além de ruído gaussiano nas medições de pressão. O método do ângulo entre vetores se destacou, apresentando eficiência de aproximadamente 98% em testes com variação de tamanho de vazamento e nos de aplicação de ruído.

Qi et al. (2014) propõem um modelo combinando algoritmos genéticos e teoria Bayesiana para detecção e localização de vazamentos. A coleta de dados foi realizada nos pontos 7, 9, 11, 14 e 16, enquanto os vazamentos simulados ocorreram nos nós 2, 3 e 15, variando de 10 L/s a 50 L/s, totalizando 15 cenários distintos. O melhor desempenho foi observado no nó 15, com um vazamento de 50 L/s, atingindo uma precisão de 94,03% na identificação do vazamento. A abordagem se destaca pela eficácia em cenários com vazamentos significativos.

O estudo conduzido por Soldevila et al. (2016) na rede de distribuição de água de Hanói usou dois sensores de pressão nos nós 14 e 30. O método proposto agrupou nós cujas assinaturas de vazamento eram semelhantes, reduzindo-os a duas classes compostas e 22 classes de nó único. Os vazamentos simulados variaram entre 25 e 75 L/s, com ruído gaussiano nas medições de pressão e incerteza nas demandas. O algoritmo atingiu uma acurácia geral de 100% na identificação de vazamentos com incerteza de vazamento ou ruído, e 89,95% sob incerteza de demanda.

Quiñones-Grueiro et al. (2018 (2)) propõem uma abordagem não supervisionada para detecção e localização de vazamentos em redes de distribuição de água. Essa abordagem é baseada em uma divisão em zonas da rede e requer apenas dados de um cenário de operação normal da rede de tubulação. A proposta combina uma transformação periódica e uma extensão de vetor de dados juntamente com a análise de componentes principais para detecção de vazamentos. Um índice de contribuição baseado em reconstrução é utilizado

para determinar a localização da zona de vazamento. A rede de distribuição de Hanói é empregada como estudo de caso para ilustrar a viabilidade da proposta. Vazamentos únicos são simulados com magnitudes de escoamento variadas em todos os nós, representando menos de 2,5% da demanda total da rede e entre 3% e 25% da demanda do nó. Todos os vazamentos podem ser detectados em um intervalo de tempo de um dia, e a taxa média de classificação obtida é de 85,28% usando apenas dados de três sensores de pressão.

Neste artigo, Quinones-Grueiro et al. (2018 (1)) discutem o uso de classificadores supervisionados para a localização de vazamentos em redes de distribuição de água. A comparação considera quatro ferramentas de classificação amplamente utilizadas no reconhecimento de padrões: K-Vizinhos Mais Próximos (Nearest Neighbor), Classificador de Bayes, Redes Neurais Artificiais e Máquinas de Vetores de Suporte (Support Vector Machines). Com intuito de aumentar a complexidade dos testes foram realizados diversos testes: o tamanho do vazamento varia conforme a fórmula do coeficiente emissor; adicionou-se um ruído de ± 5% nas medições; Variou-se o coeficiente de Hazen-Williams para simular a incerteza da rugosidade da tubulação; acrescentou-se uma incerteza na demanda de ± 10% e por fim, analisou-se a combinação destes cenários obtendo-se como melhor resultado uma acurácia de 63% para o Classificador de Bayes.

Através do monitoramento da vazão, Shravani et al. (2019) propuseram um método para identificar e localizar vazamentos em tubulações, utilizando simulações realizadas no software STAR-CCM. Diversas técnicas de aprendizado de máquina foram aplicadas com valores obtidos dessas simulações, destacando-se a rede neural multicamadas, que obteve o melhor resultado entre as analisadas, com uma acurácia de 95%.

Sun et al. (2020) apresentam um método baseado em dados para a localização de vazamentos, utilizando medições de pressão limite em RDAs. A abordagem consiste em duas etapas: (1) desenvolvimento de classificadores de aprendizado de máquina, especificamente análise discriminante linear (LDA) e redes neurais artificiais (RNA), para estimar as probabilidades de vazamento em cada nó; (2) aplicação de raciocínio temporal bayesiano para refinar essas probabilidades ao longo do tempo após a detecção de um vazamento, melhorando a precisão da localização. O método foi testado usando a área metropolitana hipotética do distrito de Hanói, alcançando mais de 80% de precisão nos melhores casos.

Por meio de simulações no EPANET de redes de distribuição de água, Fan, Zhang e Yu (2021) propôs diversas abordagens para a identificação de vazamentos. A primeira delas utilizou uma rede neural artificial, que demonstrou resultados satisfatórios em conjuntos de dados balanceados, algo não usual de conseguir para redes de distribuição reais. Desenvolveu-se também uma rede neural do tipo autoencoder, que também alcançou resultados excelentes em conjuntos de dados não balanceados dentro da área monitorada pelos sensores. No entanto, a acurácia da rede diminuiu significativamente quando os vazamentos foram simulados fora dessa área. Como solução, foi implementada uma estratégia diferente

que consiste em múltiplas tentativas de detecção, visando aumentar a acurácia do modelo e reduzir a frequência de alarmes falsos.

O estudo de (LUO et al., 2024) utiliza transformers para detectar vazamentos, o método proposto utiliza mecanismo de atenção *(attention)* para correlacionar dados históricos com dados atuais. Os dados de pressão são normalizados e realiza-se uma concatenação dos *embeddings* de posição com os valores de pressão, isso evita a perda das características originais do vazamento. A RDA de Hanói foi simulada no EPANET e gerou-se 10 cenários com diferentes tipos de vazamento (abruptos, como rupturas de canos, e outros menores). O tempo de aquisição foi fixado em 30 minutos ao longo de 12 meses. Este método alcançou uma acurácia média de 95,19% e um F1-score de 97,52%. Constatou-se também que uma janela de monitoramento de 7 dias proporciona os melhores resultados de detecção . O trabalho de (RODRÍGUEZ-ARGOTE et al., 2023) busca analisar um cenário muito mais complexo: a identificação de dois vazamentos simultâneos na rede. A abordagem junta a alocação otimizada de sensores de pressão (baseada na análise de sensibilidade e resíduos) com classificadores supervisionados de aprendizado de máquinas, testando especificamente o algoritmo KNN e a análise discriminante linear (LDA). O estudo de caso dividiu a rede de Hanói em 11 zonas (cada zona englobando pelo menos 2 nós) com taxas de vazamento variando de 0,27% a 2,7% do fluxo nominal e um ruído gaussiano de ±5% nas medições. O LDA demonstrou melhores resultados quando comparado com o KNN. A zona exata alcançou uma acurácia de aproximadamente 70%, ao aumentar a tolerância para um nó este valor foi para 82,5% e quando aumentou para 2 nós de tolerância a acurácia foi de 90,5%.

O estudo Lučin et al. (2021) demonstra a utilização de um classificador Random Forest para detectar vazamentos em redes de distribuição de água de diferentes tamanhos, com sensores espalhados pela rede. Foram simulados diversos cenários de vazamento usando parâmetros determinados por Monte Carlo, levando em conta variações diárias de demanda. A precisão do modelo foi testada em diferentes layouts de sensores espalhados pelos nós da rede. Os resultados mostram que o modelo tem maior precisão em detectar grandes vazamentos, alcançando 82% de precisão na rede de Hanói com variações menores de demanda. Mesmo para pequenos vazamentos e grandes variações de demanda, a precisão foi significativa, chegando a 89% ao considerar os cinco nós com maior probabilidade de vazamento.

Os estudos apresentados destacam a relevância da rede de Hanói como *benchmark* para métodos de detecção e localização de vazamentos. Essas pesquisas fundamentam as definições da rede a serem adotadas nesta dissertação, desta forma destaca-se o estudo de Lučin et al. (2021), que realizou uma análise abrangente variando o coeficiente de emissor de vazamento nas faixas de 1 a 5, 5 a 10, 5 a 15 e 10 a 15. Os autores observaram que coeficientes menores, especificamente no intervalo de 1 a 5, representam vazamentos

muito pequenos, dificultando a detecção desta falha. Por essa razão, os autores optaram por utilizar um coeficiente fixo igual a 10 em suas análises principais. Contudo, como este trabalho tem por objetivo investigar precisamente o comportamento de pequenos vazamentos, definiu-se a utilização de um coeficiente de emissor fixo com valor de 5. Além disso, o estudo de Lučin et al. (2021) serviu como referência para os padrões de consumo adotados. No entanto, diferentemente dessa abordagem, em vez de utilizar grupos de nós para cada padrão, criaram-se 40 novos padrões (além dos 6 originais), os quais foram atribuídos de forma aleatória a cada nó de consumo durante as simulações.

Com o intuito de facilitar a comparação entre os trabalhos relacionados, a Tabela 1 sumariza as principais características de cada um dos trabalhos mencionados anteriormente.

Tabela 1 – Quadro comparativo da literatura consultada

Autor	**AM**	**Simulação**	**EPANET**	**GNN**	**Hanói**	**Pressão**	**Resíduos**
(RASHID; AKRAM; KHAN, 2015)	✓					✓	
(LIU et al., 2019)	✓	✓					
(MASHHADI et al., 2021)	✓	✓	✓			✓	
(FAN; YU, 2022)	✓	✓	✓			✓	
(FAN; ZHANG; YU, 2021)	✓	✓	✓			✓	
(ROLLE et al., 2022a)	✓	✓	✓			✓	✓
(ROLLE et al., 2022b)	✓	✓	✓	✓		✓	✓
(ROLLE et al., 2024)	✓	✓	✓	✓		✓	✓
(RODRIGUES; ROLLE; GODOY, 2024)	✓	✓	✓	✓		✓	✓
(ROLLE, 2024)	✓	✓	✓	✓		✓	✓
(PéREZ et al., 2011)		✓				✓	✓
(PONCE; CASTAñóN; CAYUELA, 2013)		✓	✓		✓		
(QI et al., 2014)	✓	✓					
(SOLDEVILA et al., 2016)	✓	✓	✓		✓	✓	✓
(QUIñONES-GRUEIRO et al., 2018 (2))	✓	✓	✓		✓	✓	✓
(QUINONES-GRUEIRO et al., 2018 (1))	✓	✓	✓		✓	✓	✓
(SHRAVANI et al., 2019)	✓	✓					
(SUN et al., 2020)	✓	✓	✓		✓	✓	
(FAN; ZHANG; YU, 2021)	✓	✓	✓			✓	
(LUO et al., 2024)	✓	✓	✓		✓	✓	✓
(RODRÍGUEZ-ARGOTE et al., 2023)	✓	✓	✓		✓	✓	✓
(LUčIN et al., 2021)	✓	✓	✓		✓	✓	

Fonte: Elaborada pelo autor.

4 MATERIAIS E MÉTODOS

Conforme discutido nos capítulos anteriores, a detecção e localização de vazamentos em RDAs é um desafio complexo para o qual ainda não existe uma técnica amplamente estabelecida e eficaz. Diante dessa problemática, esta seção propõe uma metodologia baseada em GNNs, visando à identificação e localização precisa de vazamentos. Espera-se que, com essa abordagem, seja possível obter resultados de alta precisão, oferecendo uma solução eficiente para um problema recorrente nesse tipo de infraestrutura.

Assim, o objetivo principal deste trabalho é explorar a topologia natural das RDAs e utilizá-la como entrada para o treinamento de modelos de GNN, com o intuito de identificar e localizar vazamentos ao longo da rede. A abordagem proposta envolve a simulação de diversos cenários, utilizando o conjunto de ferramentas EPANET-MATLAB para representar variações nos padrões de consumo e simulações de vazamentos em diferentes localizações. Esses cenários geram dados cruciais, como as variações de pressão em cada vértice da rede, que são fundamentais para o treinamento do modelo GNN.

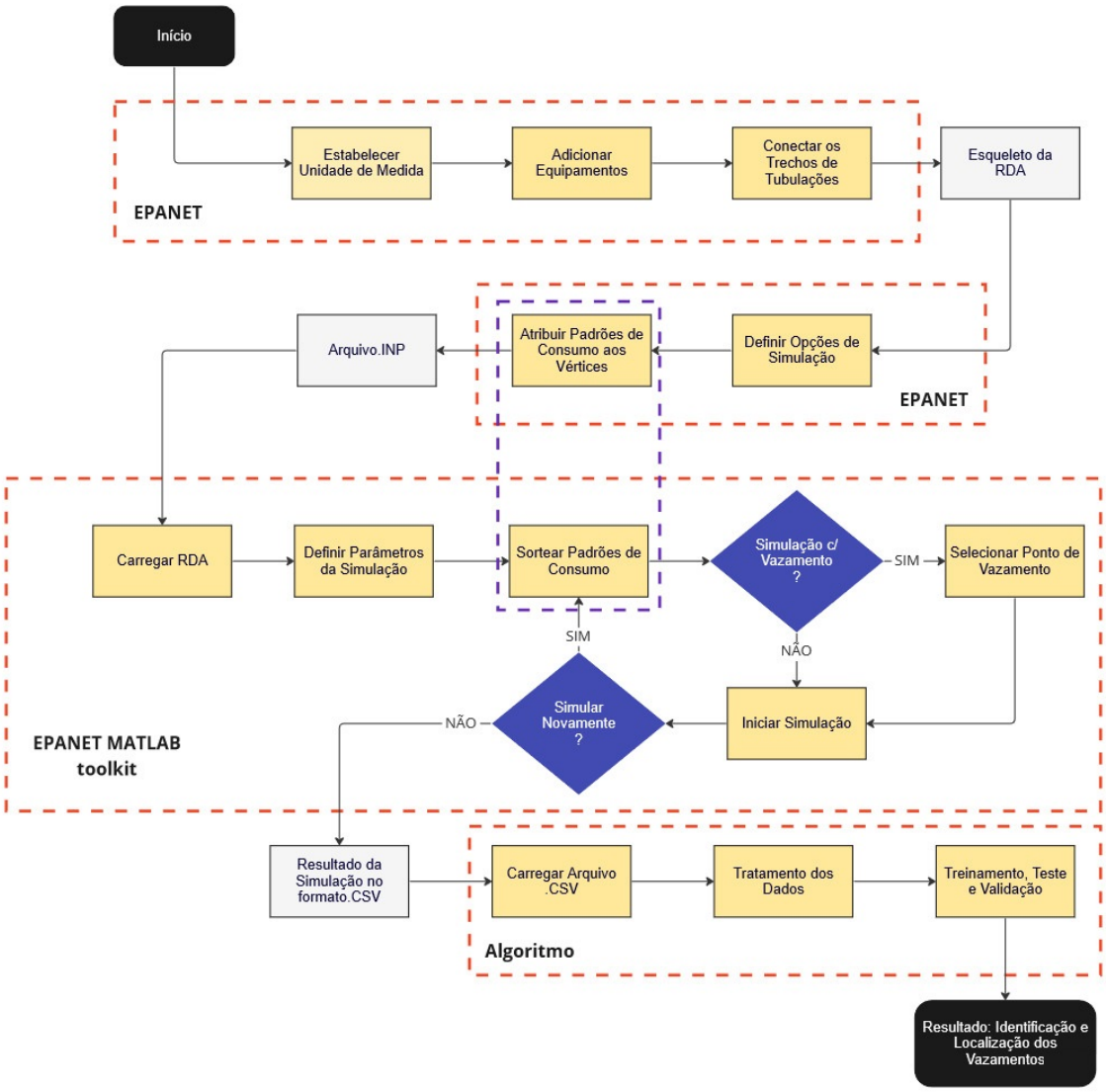
A proposta segue três etapas principais:

- **Simulação de Cenários:** Utiliza-se o conjunto EPANET-MATLAB para modelar a RDA, simulando cenários com variações no padrão de consumo e na ocorrência de vazamentos. Esses cenários refletem o comportamento hidráulico da rede em condições normais e anômalas, permitindo gerar dados essenciais, como os valores de pressão, vazão e volume nos diferentes vértices da rede.
- **Treinamento dos Modelos de GNN:** A partir dos dados simulados, os modelos de GNN são treinados para reconhecer padrões estruturais e comportamentais da RDA. A grande vantagem das GNNs está em sua capacidade de capturar as interações entre os vértices e as arestas, levando em conta a topologia da rede como um todo, o que é fundamental para a detecção de vazamentos.
- **Identificação e Localização de Vazamentos:** Os modelos, após treinados, devem ser capazes de detectar vazamentos com alta precisão e fornecer a localização das falhas dentro da RDA.

A Figura 7 apresenta de forma as etapas citadas acima, e detalhadas em seguida.

- **EPANET:**
  - **Estabelecer Unidade de Medida:** Para este trabalho, utilizaram-se as unidades do Sistema Internacional de Unidades (SI).

Figura 7 – Esquemático detalhado das etapas do projeto.



Fonte: Elaborada pelo autor.

– **Adicionar Equipamentos:** De forma manual, adicionam-se todos os equipamentos (bombas, reservatórios, etc.), inclusive os nós de consumo e os abstratos, onde serão simulados os vazamentos.

– **Conectar os Trechos de Tubulações:** Conectam-se as tubulações aos nós e equipamentos. Um ponto de atenção é a definição do comprimento do trecho, o qual deve ser adicionado manualmente. Como padrão do EPANET, de início, todos os trechos são computados com o comprimento de 1000 metros.

– **Definir Opções de Simulação:** Neste ponto, escolhem-se principalmente as opções de tempo (duração total, intervalo padrão, início das simulações, etc.).

– **Atribuir Padrões de Consumo aos Vértices:** Os padrões de consumo devem ser atribuídos manualmente a cada um dos nós de consumo.

  - **Exportar arquivo .INP:** Por fim, exporta-se um arquivo com a extensão .INP que contém todas as informações configuradas anteriormente.

- **EPANET-MATLAB:**
  - **Carregar RDA:** Carrega-se o arquivo .INP dentro do *toolkit*.
  - **Definir Parâmetros de Simulação:** Aqui, definem-se todas as especificidades da simulação, tais como: ruído aleatório, coeficiente para controle do tamanho do vazamento, duração total da simulação, etc.
  - **Sortear Padrões de Consumo:** Apesar de alguns padrões de consumo serem atribuídos no arquivo do EPANET, é necessário que, dentro do EPANET-MATLAB, programe-se a troca iterativa dos padrões atribuídos aos nós de consumo.
  - **Selecionar Ponto de Vazamento:** De forma iterativa, seleciona-se o ponto de vazamento. Após o fim da simulação referente a esse ponto, não se pode esquecer de zerar o vazamento, de forma a evitar que se tenham dois vazamentos ocorrendo simultaneamente, cenário não abordado nesta pesquisa.
  - **Iniciar Simulação:** A interação entre o MATLAB e o EPANET é necessária para que se possa simular o comportamento da rede por longos períodos, de forma ininterrupta. No final da simulação, pode-se escolher o formato de arquivo em que se deseja salvar os resultados; neste caso, escolheu-se o formato .CSV, que irá conter as seguintes informações: intervalo de tempo, valores de pressão e colunas de rótulos. Cada linha desse arquivo corresponde a uma medição com intervalo de 10 minutos.

- **Algoritmo:**
  - **Carregar arquivo .CSV:** Por meio da linguagem de programação Python e de suas bibliotecas, o arquivo .CSV é carregado.
  - **Tratamento dos Dados:** O tratamento dos dados é realizado utilizando-se as bibliotecas do Python: NumPy, Pandas e scikit-learn.
  - **Treinamento, Validação e Teste:** Com o auxílio da biblioteca scikit-learn, os dados são divididos em treino, validação e teste na proporção de 70%, 15% e 15%. Posteriormente, os modelos de redes neurais são treinados utilizando-se a biblioteca PyTorch.

A simulação das RDAs é uma etapa crucial deste trabalho, devendo ser realizada de forma a representar com fidelidade as variações hidráulicas que ocorrem em sistemas de distribuição de água. No EPANET, cada vértice de uma (RDA) pode ser configurado de acordo com as necessidades específicas do estudo, permitindo uma modelagem detalhada e

flexível. Neste estudo, será analisada a RDA da cidade de Hanói no Vietnã proposta por Fujiwara e Khang (1990). Essa rede é amplamente empregada em pesquisas de identificação e localização de vazamentos, conforme discutido na Seção 2. Na configuração da simulação, os vértices representarão pontos de consumo, onde as medições serão realizadas ou locais designados para simular vazamentos específicos. Destaca-se que os nós onde serão simulados os vazamentos não estarão representados diretamente na matriz de adjacência, uma vez que serão ativados apenas para simular o vazamento no ponto analisado.

4.1 CONSTRUÇÃO DOS MODELOS DE RDAs

Será realizado uma exploração teórica detalhada da construção de Redes de Distribuição de Água (RDA) utilizando a ferramenta EPANET.

A água é um recurso vital, não apenas para a vida, mas também para uma ampla gama de processos industriais. Essa necessidade fundamental levou ao surgimento de cidades nas proximidades de grandes rios. À medida que as populações urbanas se expandiam, tornou-se imperativo transportar água por distâncias cada vez maiores, o que motivou o desenvolvimento das primeiras tubulações para o transporte hídrico e, consequentemente, das redes de distribuição.

Com o tempo, essas redes evoluíram e se tornaram mais complexas, exigindo a instalação de uma variedade de equipamentos ao longo de sua extensão, como válvulas, bombas e reservatórios. Esse aumento na complexidade não apenas melhorou a eficiência do sistema, mas também trouxe novos desafios em termos de manutenção e operação.

Tejaswini, Silo e Ashwini (2020) define uma rede de distribuição de água (RDA) como uma simplificação de todos os componentes presentes na sua extensão integrados a equações hidráulicas de forma que se obtenha as variáveis contidas no sistema (pressão, vazão, volume, etc.)

Neste contexto, o EPANET surge como uma ferramenta crucial, onde é possível realizar simulações que ajudem a prever o comportamento das redes sob diferentes condições operacionais e a identificar áreas que necessitam de melhorias ((EPA), 2000).

Uma das limitações do EPANET é a impossibilidade de automatizar a extração de dados simulados, exigindo que esse processo seja realizado manualmente. No entanto, foi criado o EPANET-MATLAB Toolkit, que fornece uma interface programável para o EPANET através do MATLAB (ELIADES et al., 2016). A instalação da versão mais recente deste programa pode ser feita seguindo as instruções disponíveis neste link: <https://github.com/OpenWaterAnalytics/EPANET-Matlab-Toolkit>

Os modelos hidráulicos desempenham um papel crucial na análise de redes de distribuição de água, principalmente devido às dificuldades práticas envolvidas na realização

de testes em sistemas reais. Essas redes frequentemente se estendem por centenas de quilômetros, e a implementação de testes diretos pode comprometer significativamente o fornecimento de água a um grande número de consumidores. Nesse contexto, a modelagem hidráulica emerge como uma solução valiosa, permitindo a simulação de uma variedade de cenários operacionais sem a necessidade de intervenções na infraestrutura existente.

Essa abordagem não apenas possibilita a avaliação de diferentes condições operacionais, mas também a identificação de problemas potenciais, como vazamentos, sem prejudicar o serviço prestado à população.

De maneira simplificada, uma RDA pode ser representada como um grafo, onde os vértices correspondem aos equipamentos da rede (como bombas e válvulas) e as arestas representam as tubulações responsáveis pelo transporte de água, desde a fonte até o consumidor final.

Baseado no conceito de grafos, a modelagem de uma RDA é realizada conforme detalhado a seguir: ((EPA), 2000)

1. **Desenho da Representação Esquemática:** O primeiro passo envolve criar uma representação esquemática simplificada da RDA (grafo) ou importar uma descrição-base do sistema a partir de um arquivo de texto.

2. **Edição de Propriedades:** Em seguida, é necessário editar as propriedades dos objetos que compõem o sistema, definindo características essenciais para a simulação.

3. **Descrição das Condições Operacionais:** Esta fase envolve a descrição detalhada das condições de operação do sistema, garantindo que todos os parâmetros relevantes sejam considerados.

4. **Seleção de Opções de Simulação:** Os usuários devem escolher as opções de simulação que desejam realizar, seja em relação ao comportamento hidráulico ou à qualidade da água.

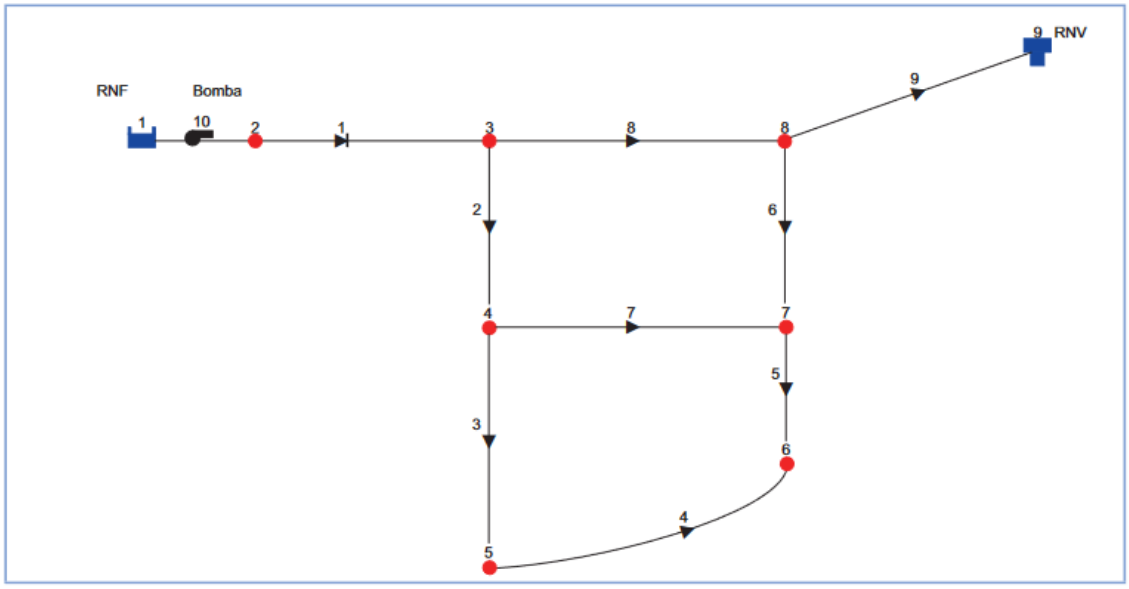
   Neste contexto, ressalta-se a importância de configurar adequadamente as opções de tempo, com ênfase no no passo de tempo do relatório (*reporting time step*). Por definição padrão do EPANET, este parâmetro assume o valor de uma hora, o que implica que os parâmetros da rede serão atualizados apenas ao final de cada ciclo de 60 minutos, independentemente das variações hidráulicas que ocorram neste período. Caso este valor não seja ajustado para intervalos menores, tentativas de aquisição de dados em frequências mais altas (como a cada 10 minutos) resultarão em medições redundantes e repetidas dentro de uma mesma janela horária. Essa redundância pode prejudicar o treinamento de algoritmos de aprendizado de máquina, uma vez que o modelo não receberá informações sobre a dinâmica da rede.

5. **Execução das Simulações:** Após a configuração, as simulações são executadas, permitindo a análise do desempenho do sistema sob diferentes condições.

6. **Visualização e Salvamento dos Resultados:** Por fim, os resultados obtidos podem ser visualizados e salvos para análises posteriores.

A Figura 8 ilustra um exemplo de uma RDA simples desenhada no EPANET. Neste exemplo, a rede é composta por um reservatório de nível fixo (RNF), que mantém um nível de água constante, e um reservatório de nível variável (RNV). As características de cada vértice (componentes) e aresta (tubulações) são definidas pelo usuário, permitindo um controle refinado sobre a simulação

Figura 8 – Representação de uma RDA desenhada no EPANET.



Fonte: (EPA) (2000)

4.1.1 MODELAGEM DO CONSUMO DE ÁGUA

A modelagem do consumo de água é uma etapa crucial na simulação de sistemas de distribuição, pois permite prever as demandas de água em diversos pontos da rede. Através dessas simulações, é possível dimensionar adequadamente projetos de infraestrutura e identificar distorções, facilitando a detecção de falhas potenciais. No EPANET, a modelagem do consumo de água pode ser realizada de duas maneiras distintas: **consumo constante** e por **padrões de consumo**.

O **consumo constante** refere-se a uma abordagem onde a demanda de água em um determinado ponto da rede é considerada fixa ao longo do tempo. Essa modelagem é particularmente útil em cenários onde o padrão de uso da água não varia significativamente,

como em indústrias que operam em turnos regulares ou em áreas residenciais onde o consumo é previsível.

Utilizar o consumo constante simplifica as simulações, permitindo que se analise o desempenho da rede em condições estáveis. Contudo, essa abordagem pode não capturar variações sazonais ou diárias, levando a uma subestimação ou superestimação das necessidades reais em certos períodos.

Por outro lado, a modelagem por **padrões de consumo** permite uma representação mais dinâmica da demanda de água. No EPANET, isso é feito utilizando dados históricos e estatísticas para estabelecer perfis de consumo que refletem as variações temporais, como horários de pico, dias da semana e estações do ano. Essa abordagem é especialmente importante em áreas urbanas, onde a demanda pode flutuar significativamente ao longo do dia e em resposta a eventos especiais.

A utilização de padrões de consumo proporciona uma análise mais precisa do sistema de distribuição, facilitando o dimensionamento adequado das infraestruturas e a identificação de possíveis falhas durante períodos de alta demanda.

Os padrões de consumo variam para cada indivíduo, dependendo das características específicas de cada consumidor, seja ele residencial, industrial, rural ou de outra natureza. Assim, dentro de uma mesma RDA, é comum encontrar diferentes padrões de consumo, refletindo as necessidades distintas de cada um.

Na Figura 9 é possível observar um exemplo de padrão de consumo ao longo de um intervalo de 24 horas, onde cada período de consumo é representado em intervalos de 3 horas e o fator multiplicativo associado a cada período indica o volume de água consumido durante esse intervalo específico.

O consumo de água em cada vértice da rede, em um dado instante, é determinado pelo produto do consumo base desse vértice e o fator multiplicativo do padrão de consumo correspondente ao momento (ROSSMAN et al., 2020).

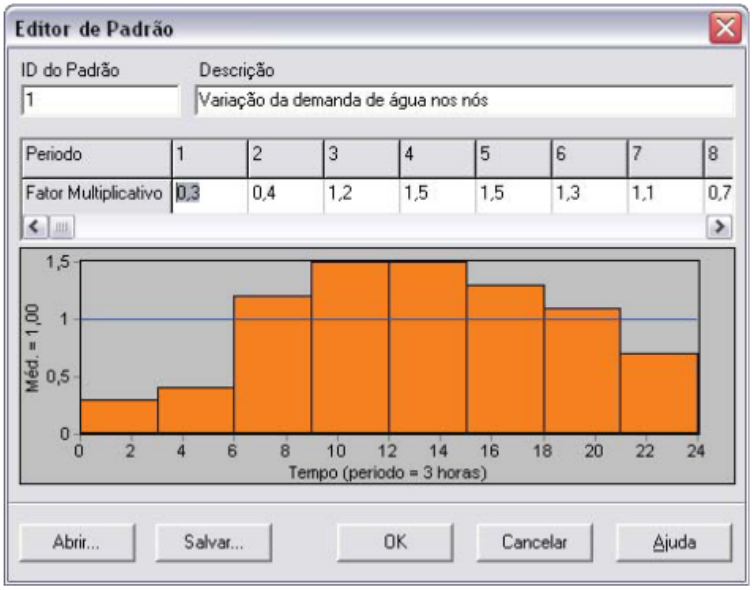
A demanda de cada nó dada pela equação 4.1:

$$d_i(k) = bd_i \cdot pa_i(k) \tag{4.1}$$

Onde bd_i é o valor de consumo base, e $pa_i(k)$ é o valor do padrão associado ao nó i no instante de tempo k (WALSKI et al., 2003).

Em síntese, a modelagem do consumo é um aspecto indispensável para a realização de simulações hidráulicas no EPANET. Portanto, é fundamental que essa modelagem seja realizada de maneira cuidadosa, assegurando que os padrões de consumo representados sejam o mais próximo possível da realidade dos consumidores.

Figura 9 – Padrão de consumo para um período de 24 horas.



Fonte: Elaborada pelo autor.

4.1.2 MODELAGEM DOS VAZAMENTOS DE ÁGUA

Os vazamentos em RDAs representam um problema crítico, tanto em termos de desperdício de recursos quanto de prejuízos financeiros e impactos ambientais. Esses vazamentos podem ser classificados em duas categorias principais: **a)** rompimentos abruptos e **b)** vazamentos pequenos causados por fraturas ou furos nas tubulações.

Os rompimentos abruptos, frequentemente chamados de *burst*, são relativamente mais fáceis de detectar devido à magnitude das alterações que provocam. Eles causam mudanças significativas na dinâmica hidráulica da rede, como quedas bruscas de pressão, e podem gerar sinais visíveis na superfície, como poças d'água, rachaduras ou movimentação de solo. Essas características tornam sua identificação mais direta e rápida.

Por outro lado, os vazamentos menores, oriundos de pequenas fraturas ou furos, apresentam um desafio muito maior. Geralmente, esses vazamentos não causam mudanças perceptíveis no sistema e não geram sinais visíveis na superfície, exigindo, assim, técnicas mais avançadas para sua detecção.

Os vazamentos em RDAs são simulados através da inserção de nós específicos para este fim, e dessa forma não aparecem na estrutura do grafo. Nesse modelo, os vazamentos não são tratados como consumos constantes. Em vez disso, utilizam-se coeficientes dinâmicos que relacionam o vazamento à pressão existente no nó.

A equação utilizada para modelar a vazão de um vazamento é dada por:

$$q_l = C_e p^{\gamma} \tag{4.2}$$

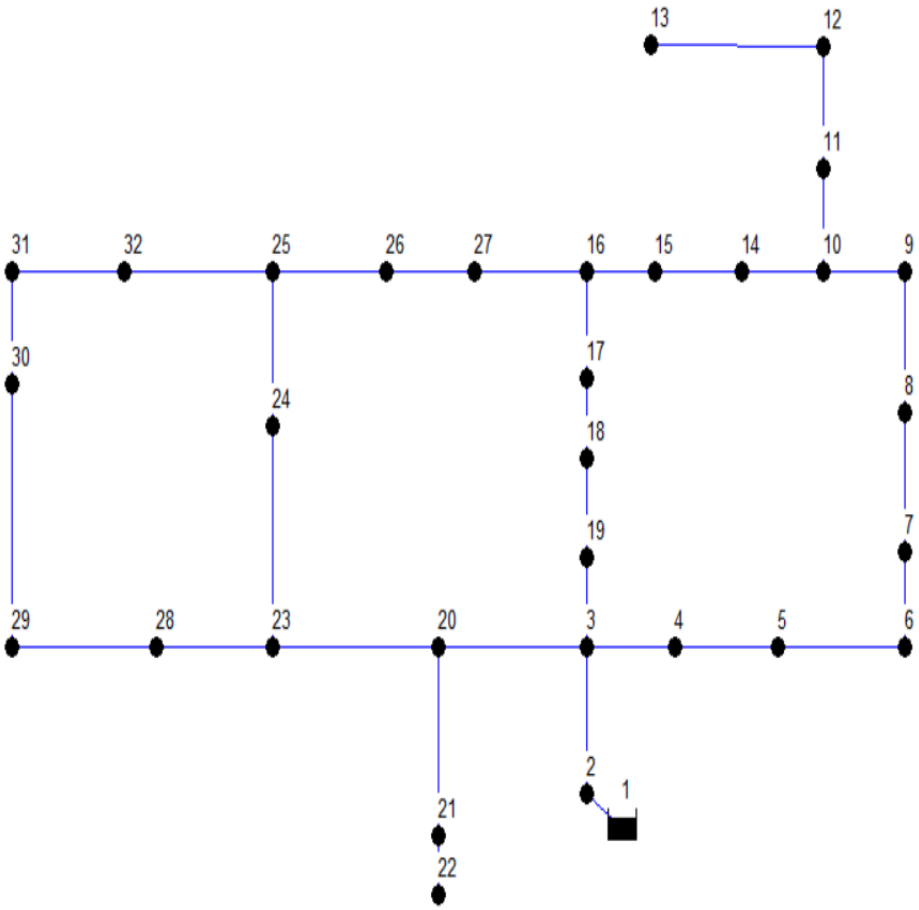
Onde q_l é a vazão do vazamento, C_e é o coeficiente de emissão que reflete o tamanho do vazamento, p é a pressão no nó, e γ é um expoente que descreve como o fluxo do vazamento varia com a pressão, este valor depende do material e da condição da tubulação

Todo o processo de simulação das RDAs está de acordo com o artigo (TOMAZINI et al., 2024) qual detalha passo a passo desde a configuração do EPANET até o arquivo final com os dados extraidos das simulações.

4.2 RDA DA CIDADE DE HANÓI NO VIETNÃ

A RDA da cidade de Hanói foi proposta por Fujiwara e Khang (1990), a topologia desta rede está demonstrada na Figura 10.

Figura 10 – RDA da cidade de Hanói.



Fonte: Elaborado pelo autor.

A rede de Hanói é composta por um conjunto de vértices interligados por tubulações,

formando um sistema que simula a distribuição de água em um ambiente urbano real. Ela é constituída por 32 vértices, que representam conglomerados de consumidores (residências, indústrias ou outros estabelecimentos) distribuídos ao longo de toda sua extensão, sendo que o vértice 1 representa a saída do reservatório de nível fixo. Todos os nós encontram-se localizados em uma elevação de zero metros, ou seja, não há influência da altura dos vértices na variação de pressão da rede e a pressão mínima em cada vértice de consumo deve ser de 30 mca.

Esses 32 vértices são conectados por 34 tubulações com comprimentos variados e resistências hidráulicas específicas, fatores que influenciam diretamente na perda de carga (pressão) e no comportamento do fluxo de água. Essa variabilidade aumenta a complexidade na previsão do comportamento hidráulico da RDA, tornando a detecção de vazamentos uma tarefa mais desafiadora.

A rede também inclui um reservatório de nível fixo que abastece todos os vértices de consumo. É importante ressaltar que não há uma bomba impulsionando a água na rede. O abastecimento é feito exclusivamente pela diferença de altura (gravidade), o que influencia o comportamento do sistema e deve ser considerado na análise dos fluxos e pressões.

A distância e comprimento de cada trecho está sumarizada na tabela 2

4.2.1 CÁLCULO DOS RESÍDUOS

Esta etapa converte as medições de pressão da RDA em valores residuais de forma a isolar algumas variações e facilitar a convergência do modelo analisado. Em vez de utilizar os valores absolutos de pressão lidos pelos sensores, é realizado um processo que recebe como entrada o valor residual de cada sensor em relação ao valor medido pelo sensor na entrada da RDA.

Para cada sensor n em um instante de tempo t, o valor de entrada P_n é transformado em um resíduo por meio da subtração de uma pressão de referência (P_{ref}) que normalmente é a entrada da rede.

$$P_n^{'} = P_n - P_{ref}$$

Onde, P_n é o valor de pressão lido pelo sensor, P_{ref} é o valor de referência que é o nó de entrada e $P_n^{'}$ é o valor residual da pressão.

4.2.2 PRÉ-PROCESSAMENTO DOS DADOS

Antes de iniciar o treinamento da rede neural com os dados da RDA de Hanói, foi aplicado um processo de normalização conforme equação 4.3. Esse procedimento é

Tabela 2 – Tabela de tubulações, nós e dimensões da RDA da cidade de Hanói

Tubulação	**Nó_i**	**Nó_o**	**Comprimento (m)**	**Diâmetro (mm)**
1	1	2	100,00	1016,00
2	2	3	1350,00	1016,00
3	3	4	900,00	1016,00
4	4	5	1150,00	1016,00
5	5	6	1450,00	1016,00
6	6	7	450,00	1016,00
7	7	8	850,00	1016,00
8	8	9	850,00	1016,00
9	9	10	800,00	1016,00
10	10	11	950,00	762,00
11	11	12	1200,00	609,60
12	12	13	3500,00	609,60
13	10	14	800,00	508,00
14	14	15	500,00	304,80
15	15	16	550,00	304,80
16	16	17	2730,00	304,80
17	17	18	1750,00	508,00
18	18	19	800,00	508,00
19	19	3	400,00	609,60
20	3	20	2200,00	1016,00
21	20	21	1500,00	508,00
22	21	22	500,00	304,80
23	20	23	2650,00	1016,00
24	23	24	1230,00	762,00
25	24	25	1300,00	762,00
26	25	26	850,00	508,00
27	26	27	300,00	304,80
28	27	16	750,00	304,80
29	23	28	1500,00	406,40
30	28	29	2000,00	304,80
31	29	30	1600,00	304,80
32	30	31	150,00	508,00
33	31	32	860,00	406,40
34	32	25	950,00	609,60

Fonte: Elaborada pelo autor.

essencial para garantir que os dados de entrada estejam em uma escala adequada para a rede neural, o que facilita o processo de aprendizado. O *MinMaxScaler* transforma os dados para um intervalo definido (GéRON, 2017). No contexto deste trabalho, escolheu-se o intervalo entre 0 e 1.

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}, \tag{4.3}$$

onde, $\mathbf{X}$ é o valor original, $\mathbf{X_{min}}$ e $\mathbf{X_{max}}$ são os valores mínimos e máximos dos dados, respectivamente.

Esse tipo de normalização é particularmente importante para algoritmos de aprendizado de máquina que são sensíveis às magnitudes das entradas, como redes neurais, pois ajuda a evitar que características com valores maiores dominem o processo de treinamento, prejudicando a eficiência do modelo. Além disso, ao uniformizar a escala dos dados, o treinamento tende a ser mais estável e convergir de forma mais rápida, melhorando o desempenho final da rede neural.

4.3 IMPLEMENTAÇÃO DAS REDES NEURAIS

Neste estudo todas as redes neurais analisadas foram construídas utilizando Python em conjunto com o *framework* Pytorch. A seguir será detalhada a implementação de cada uma das arquiteturas, assim como a explicação de seus respectivos algoritmos:

4.3.1 GGNN

Como visto na seção 2, este modelo utiliza unidades recorrentes do tipo GRU para atualizar os estados ocultos dos nós do grafo ao longo de um número pré-definido de passos de propagação (*Windowns size*). O processo de propagação da mensagem (*message passing*) inicia com uma transformação linear dos atributos (pressões) iniciais dos nós e em seguida por uma função de ativação ReLU e uma camada DropOut para mitigação de *overfitting*. Para cada passo de propagação, as mensagens são agregadas de duas direções: entrada (A_in) e saída (A_out), permitindo que o nó compreenda seu contexto estrutural bidirecional.

Posteriormente, os dados são passados juntos com a matriz de adjacências para GGNN, responsável por realizar a agregação dos dados da vizinhança e atualização dos estados de cada nó da rede. Após a etapa de propagação o estado oculto final passa por uma camada linear de saída e por fim por uma função softmax com intuito de prever se há vazamento e caso sim, em qual nó.

A lógica descrita no Pseudoalgoritmo 1 ilustra essa implementação. Logo de inicio calcula-se uma máscara, está mascará é necessária para que o algorimo consiga ajustar os

dados ao tamanho do *Batch size*. O laço de repetição principal (linhas 5 a 10) executa a propagação em T passos. Em cada iteração, as mensagens de entrada e saída da vizinhança são calculadas (linhas 6 e 7) e concatenadas (linha 8). Essa informação conjunta alimenta a célula GRU (GRUCell), que atualiza o estado oculto daquele nó (linha 9). Por fim, utiliza-se o LogSoftmax para gerar as probabilidades de classificação.

Algorithm 1 Gated Graph Neural Network (GGNN)

Require: Matriz de Pressões X, Matriz de Adjacência A, Número de passos T

1: $M \leftarrow (X_{:,:,0} \neq 0) \times 1$ ▷ Cálculo da máscara a partir dos atributos
2: $H^{(0)} \leftarrow \text{ReLU}(\text{Dropout}(XW_{in}))$ ▷ Projeção inicial com Dropout
3: $A_{out} \leftarrow A$ ▷ Matriz para mensagens de entrada
4: $A_{in} \leftarrow A^{\top}$ ▷ Matriz para mensagens de saída
5: **for** $t = 1$ **to** T **do**
6: $\quad a_{in} \leftarrow AH^{(t-1)}$ ▷ Agregação das mensagens de entrada
7: $\quad a_{out} \leftarrow A_{out}H^{(t-1)}$ ▷ Agregação das mensagens de saída
8: $\quad m^{(t)} \leftarrow [a_{in} \parallel a_{out}]$ ▷ Concatenação ($\parallel$)
9: $\quad H^{(t)} \leftarrow \text{GRUCell}(m^{(t)}, H^{(t-1)})$ ▷ Atualização do estado oculto
10: **end for**
11: $Z \leftarrow H^{(T)}W_{out}$ ▷ Projeção para dimensão de saída
12: $O \leftarrow \text{LogSoftmax}(Z + \log(M + \epsilon))$ ▷ Aplicação da máscara e ativação final
13: **return** O

4.3.2 G-LSTM

Esta implementação propõe uma variação da arquitetura GGNN, substituindo o mecanismo GRU padrão por uma célula LSTM. A inicialização das características dos nós assemelha-se à da GGNN: os dados de pressão de entrada passam por uma transformação linear, seguida por uma etapa de regularização *Dropout* e uma função de ativação ReLU, compondo assim o estado oculto inicial ($H^{(0)}$). Como a célula LSTM possui uma porta de memória adicional (*cell state*), os estados desta porta ($C^{(0)}$) são estritamente inicializados com zeros.

Durante o processo de propagação de mensagens (*message passing*), a rede utiliza a matriz de adjacências para calcular simultaneamente as informações das conexões de entrada (A_{in}) e de saída (A_{out}), obtida pela matriz transposta. Essas agregações vizinhas são concatenadas e atuam como a entrada da célula LSTM a cada iteração. Esse processo é repetido por um número predefinido de T passos (*windows size*). Ao final das iterações, o estado oculto resultante ($H^{(T)}$) é submetido a uma projeção linear final (W_{out}). Por fim, aplica-se uma máscara a partir das características originais para filtrar nós inválidos, seguida pela função de ativação *LogSoftmax*, para a classificação do nó.

O Pseudoalgoritmo 2 demonstra a adaptação realizada no GGNN para que seja possivel utilizar O LSTM em vez da GRU.

Algorithm 2 Graph Long Short-Term Memory (G-LSTM)

Require: Matriz de atributos dos nós X, Matriz de Adjacência A, Número de passos T

1: $M \leftarrow (X_{:,:,0} \neq 0) \times 1$ ▷ Cálculo da máscara a partir dos atributos
2: $H^{(0)} \leftarrow \text{ReLU}(\text{Dropout}(XW_{in}))$ ▷ Projeção inicial das características
3: $C^{(0)} \leftarrow \mathbf{0}$ ▷ Inicialização do estado da célula com zeros
4: $A_{in} \leftarrow A$ ▷ Matriz de adjacência para conexões de entrada
5: $A_{out} \leftarrow A^{\top}$ ▷ Matriz de adjacência para conexões de saída
6: **for** $t = 1$ **to** T **do**
7: $\quad a_{in} \leftarrow A_{in}H^{(t-1)}$
8: $\quad a_{out} \leftarrow A_{out}H^{(t-1)}$
9: $\quad m^{(t)} \leftarrow [a_{in} \parallel a_{out}]$ ▷ Concatenação das mensagens
10: $\quad H^{(t)}, C^{(t)} \leftarrow \text{LSTMCell}(m^{(t)}, H^{(t-1)}, C^{(t-1)})$ ▷ Atualização da LSTM
11: **end for**
12: $Z \leftarrow H^{(T)}W_{out}$
13: $O \leftarrow \text{LogSoftmax}(Z + \log(M + \epsilon))$
14: **return** O

4.3.3 GraphSAGE

A arquitetura GraphSAGE é baseada em camadas (K), em que as representações são geradas por meio da agregação de informações locais. Diferentemente dos modelos baseados em recorrência temporal (como GGNN e G-LSTM), o GraphSAGE atualiza o estado (*embedding*) de um nó combinando suas próprias características da camada anterior com as características processadas exclusivamente da vizinhança amostrada.

No presente trabalho, implementaram-se três funções distintas de agregação: Média (*Mean*), *Pooling* e LSTM. O Pseudoalgoritmo 3 descreve a implementação desta arquitetura, onde o laço principal é definido pelo número de camadas K. Na linha 4, a rede captura as informações da vizinhança ao multiplicar o estado atual pela matriz de adjacência. Na linha 5, a função AGGREGATE atualiza as informações do nó integrando-as à sua vizinhança, sendo esta a etapa em que se define a variação do modelo (*Pooling*, LSTM ou Média). Por fim, de forma similar às arquiteturas anteriores, a função LogSoftmax é responsável por realizar a classificação do nó.

A principal diferença estrutural entre o GraphSAGE e as arquiteturas GGNN e G-LSTM reside na mecânica de propagação da informação. Enquanto a GGNN e a G-LSTM utilizam mecanismos de recorrência temporal, aplicando os mesmos pesos iterativamente ao longo de T passos para atualizar o estado global do grafo, o GraphSAGE adota uma

abordagem espacial. No GraphSAGE, a agregação é realizada de forma hierárquica e local. Essa característica confere ao modelo uma natureza indutiva, tornando-o altamente eficiente para generalizar padrões topológicos a partir de vizinhanças limitadas, sem a necessidade de reprocessar a dinâmica global da rede a cada iteração.

Algorithm 3 GraphSAGE (Sample and Aggregate)

Require: Matriz de atributos dos nós X, Matriz de Adjacência A, Número de camadas K, Número de vizinhos a amostrar S_k

$M \leftarrow (X_{:,:,0} \neq 0) \times 1$ ▷ Cálculo da máscara a partir dos atributos

$H^{(0)} \leftarrow \text{ReLU}(XW_{in})$ ▷ Projeção inicial dos atributos

for $k = 1$ **to** K **do**

  $\tilde{A}^{(k)} \leftarrow \text{AmostraVizinhos}(A, S_k)$ ▷ Amostragem dos vizinhos

  $H_N^{(k)} \leftarrow \tilde{A}^{(k)} H^{(k-1)}$ ▷ Multiplicação pela matriz amostrada

  $H^{(k)} \leftarrow \text{AGGREGATE}_k(H^{(k-1)}, H_N^{(k)})$ ▷ Mean, Pooling ou LSTM

end for

$Z \leftarrow H^{(K)} W_{out}$ ▷ Camada linear de saída

$O \leftarrow \text{LogSoftmax}(Z + \log(M + \epsilon))$

return O

4.4 MÉTRICAS DE AVALIAÇÃO

As métricas de avaliação desempenham um papel crucial na avaliação do desempenho de modelos de aprendizado de máquina em problemas de classificação. Essas métricas fornecem uma maneira sistemática de medir e quantificar a qualidade das previsões feitas pelo modelo em relação aos rótulos reais dos dados.

Uma das ferramentas mais importantes neste processo é a matriz de confusão, que apresenta a contagem de instâncias classificadas correta e incorretamente para cada classe. A matriz de confusão é uma tabela cruzada que registra o número de ocorrências entre a classificação verdadeira (real) e a classificação prevista pelo modelo, conforme ilustrado na Figura 11 (GRANDINI; BAGLI; VISANI, 2020).

A partir da matriz de confusão, definem-se os conceitos fundamentais: Verdadeiros Positivos (VP), Falsos Positivos (FP), Verdadeiros Negativos (VN) e Falsos Negativos (FN). Com base nestes valores, é possível calcular as seguintes métricas principais:

- **Acurácia**: Mede a proporção de previsões corretas em relação ao total de previsões feitas pelo modelo. É uma métrica direta e fácil de interpretar, sendo amplamente utilizada para avaliar a eficácia geral do modelo:

Figura 11 – Exemplo de uma matriz de confusão.

		PREDICTED classification				
	Classes	a	b	c	d	Total
ACTUAL classification	a	6	0	1	2	9
	b	3	9	1	1	14
	c	1	0	10	2	13
	d	1	2	1	12	16
	Total	11	11	13	17	**52**

Fonte: Adaptado de Grandini, Bagli e Visani (2020).

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN}$$

No entanto, é importante ter cautela ao interpretar a acurácia em problemas com classes desbalanceadas, pois um modelo pode obter uma alta taxa de acerto simplesmente prevendo a classe majoritária em todos os casos (GéRON, 2017).

- **Precisão (*Precision*)**: Mede a proporção de instâncias corretamente classificadas como positivas em relação ao total de previsões positivas feitas pelo modelo. Ela fornece informações cruciais sobre a capacidade do algoritmo de evitar falsos positivos (GRANDINI; BAGLI; VISANI, 2020):

$$\text{Precisão} = \frac{VP}{VP + FP}$$

- **Sensibilidade (*Recall*)**: Mede a proporção de previsões corretamente positivas em relação ao total de instâncias que são de fato positivas no conjunto de dados. É útil para avaliar a capacidade do modelo de encontrar e identificar corretamente as anomalias ou alvos reais (HOSSIN; SULAIMAN, 2015):

$$\text{Recall} = \frac{VP}{VP + FN}$$

Geralmente, há um balanceamento (*trade-off*) entre precisão e *recall*, sendo necessário encontrar um equilíbrio que depende do problema abordado.

- ***F1-score***: Esta métrica representa a média harmônica entre a precisão e o *recall*. É amplamente utilizada quando se deseja um equilíbrio robusto entre as duas métricas, penalizando modelos que priorizam uma em detrimento da outra:

$$F1 = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}$$

É crucial selecionar as métricas de avaliação adequadas com base nas características específicas do problema e nos requisitos do contexto. Uma única métrica não é suficiente para avaliar completamente o desempenho de um modelo, e é recomendado usar uma combinação de métricas para obter uma visão mais abrangente.

4.5 PROCEDIMENTO EXPERIMENTAL

Foram elaborados 5 experimentos com intuito de validar as redes neurais analisadas para compreender suas caracteristicas e identificar suas vantagens e limitações. As simulações foram realizadas conforme os procedimentos descritos em Tomazini et al. (2024).

4.5.1 SIMULAÇÕES NO EPANET-MATLAB

Como demonstrado anteriormente, escolheu-se a rede da cidade de Hanói proposta por Fujiwara e Khang (1990). Está RDA é considerado um *Benchmark* da área de identificação e localização de vazamentos.

Para simular as variações que ocorrem em condições reais de operação, foi adicionado um ruído aleatório de aproximadamente 5% em todas as medições de pressão. Esse ruído é essencial para aproximar o modelo teórico da realidade, uma vez que, em sistemas reais, as medições de pressão raramente são totalmente precisas devido a flutuações de demanda, falhas nos sensores, condições atmosféricas e outras interferências.

A introdução de ruído também confere maior robustez ao modelo, tornando-o mais resiliente a variações inesperadas. Modelos treinados com dados perfeitos podem apresentar desempenho idealizado, mas muitas vezes falham em cenários práticos. A inclusão de ruído nas medições busca garantir que o sistema seja capaz de lidar com as imperfeições do mundo real, o que é crucial para a confiabilidade das técnicas de detecção e localização de vazamentos mesmo em condições adversas.

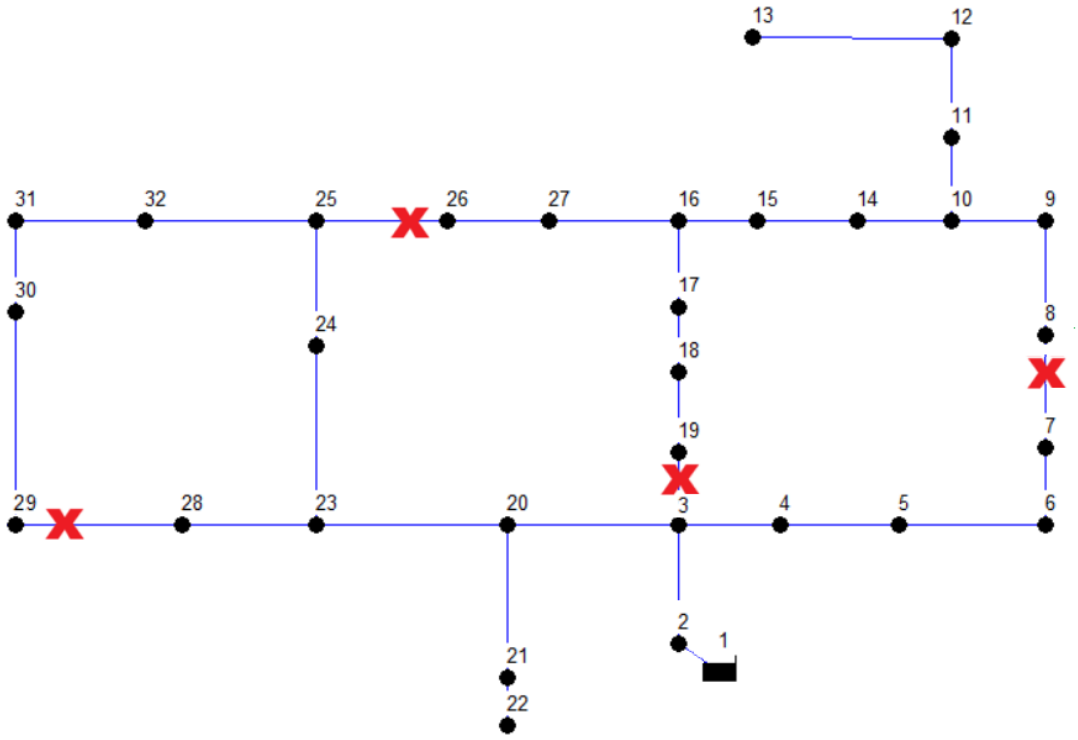
A determinação precisa da rugosidade das tubulações em sistemas de distribuição de água é desafiadora, pois essa característica não pode ser diretamente mensurada e tende a se alterar ao longo do tempo devido a fatores como corrosão, acúmulo de incrustações e deterioração dos materiais. A rugosidade impacta diretamente a eficiência do fluxo de água, afetando a perda de carga no sistema. Devido à variação com o tempo e às condições de operação, utilizam-se aproximações com base em modelos empíricos.

Conforme demonstrado na seção 4, uma das formas de estimar para estimar a rugosidade da tubulação é através do **coeficiente de Hazen-Williams**. Em consonância com o utilizado por Quinones-Grueiro et al. (2018 (1)). Salienta-se que este coeficiente é configurado diretamente no EPANET.

Os pontos onde foram simulados os vazamentos estão representados na Figura 12. Todos os vazamentos foram simulados nas tubulações, desta forma, estes pontos adicionais não aparecem diretamente na matriz de adjacências.

Como o local onde ocorre o vazamento não está presente na RDA, optou-se por **classificar o nó mais próximo ao local onde está ocorrendo o vazamento**, desta forma o modelo referencia: 0 (zero) para quando não há vazamento e os nós 8, 19, 26 e 29 quando há vazamento próximo destes, conforme demonstrado na Figura 12.

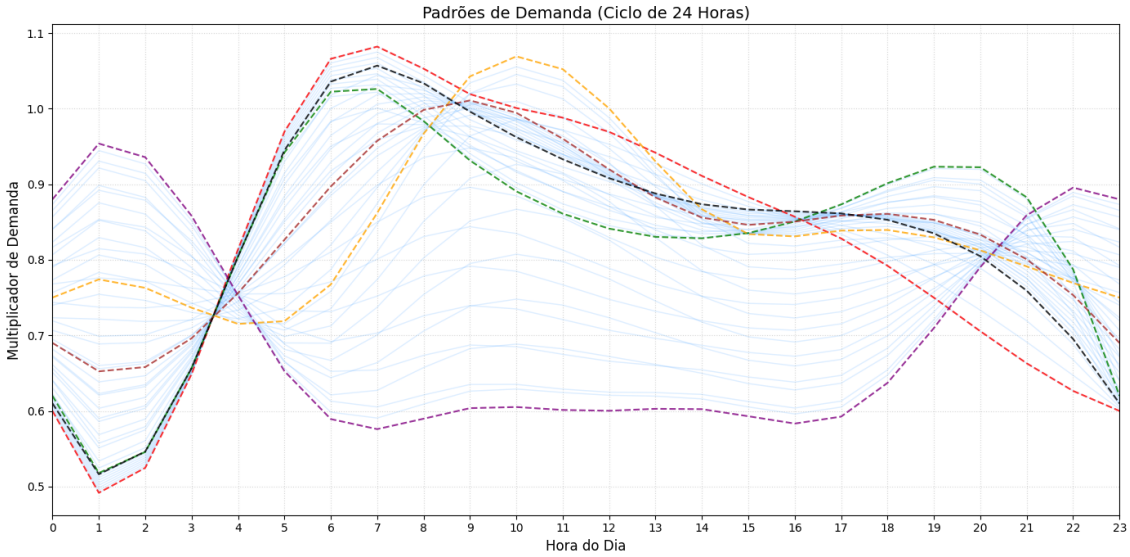
Figura 12 – Pontos de simulação de vazamentos.



Fonte: Elaborada pelo autor.

Como descrito na Seção 4, utilizou-se padrões de demanda para modelar o consumo dos nós. A Figura 13 apresenta o conjunto de padrões adotados neste estudo. Ao todo, foram elaborados 46 padrões distintos, aplicados individual e aleatoriamente a cada nó da rede. Ao analisar a figura, nota-se que alguns padrões possuem cores diferentes. Estes 6 padrões são os empregados por (LUčIN et al., 2021) e serviram como base para elaborar outros 40 padrões utilizados neste trabalho.

Figura 13 – Padrões de consumo.



Fonte: Elaborada pelo autor.

Para cada simulação, a frequência de medição foi definida em intervalos de 10 minutos, refletindo as capacidades de tecnologias de monitoramento disponíveis no mercado.

Foram realizadas **24000** simulações para cada um dos casos com incidência de vazamento e **48000** simulações com a rede em operação normal, sem vazamentos, totalizando **144000** dados simulados **(4 x 24000 nós c/ vazamento + 1 x 48000 s/ vazamento),** o que é equivalente a 1000 dias de simulação.

4.5.2 TREINAMENTO, VALIDAÇÃO E TESTE

Antes de se realizar a normalização utilizou-se a abordagem baseada em resíduos como visto na seção 4. Em seguida, aplicou-se o *MinMaxScaler* com o intuito de garantir que os dados de entrada estejam em uma escala adequada para a rede neural, que no contexto desse trabalho é de 0 a 1.

Posteriormente, esses dados foram usados para treinar, validar e testar os modelos na proporção de 70% para treinamento, 15% para validação e 15% para teste.

Para garantir a imparcialidade na comparação entre as diferentes arquiteturas, realizou-se um processo de otimização dos hiperparâmetros para cada um dos modelos analisados através da técnica de busca aleatória (*Random Search*). Essa abordagem foi escolhida devido à sua capacidade de explorar um amplo espaço amostral com um custo computacional significativamente mais baixo quando comparada a técnicas exaustivas, como o *Grid Search*. O processo de otimização foi guiado de forma automatizada visando a maximização da métrica *F1-score*. A tabela 3 demonstra os valores utilizados durante o

processo de otimização.

Tabela 3 – Espaço de busca dos hiperparâmetros

Hiperparâmetro	**Valores Testados**
Taxa de Aprendizado (*Learning Rate*)	0,01; 0,001; 0,0001
Tamanho do Lote (*Batch Size*)	100, 300, 512, 1024, 2048
Dimensão Oculta (*Hidden Dimension*)	32, 64, 128, 256, 512, 1024
Tamanho da Janela (*Window Size*)	3, 5, 10, 15

Durante a execução, limitou-se o escopo de testes a um total de 30 combinações por modelo analisado. Ao final do processo exploratório, os conjuntos de hiperparâmetros que apresentaram o melhor desempenho foram consolidados e estão apresentados na Tabela 4.

Tabela 4 – Configuração final dos hiperparâmetros otimizados por modelo

Modelo	**Taxa de Aprendizado**	**Tamanho do Lote**	**Dimensão Oculta**	**Tamanho da Janela**
G-LSTM	0,0001	300	256	10
GraphSAGE-LSTM	0,0001	300	256	3
GraphSAGE-Média	0,0001	300	256	10
GGNN	0,0001	300	256	5
GraphSAGE-Pool	0,001	1024	512	15

4.6 EXPERIMENTO 1 - ANÁLISE SOB CONDIÇÃO IDEAL

Com o intuito de avaliar o cenário em que os modelos têm acesso à totalidade dos dados disponíveis, realizou-se o treinamento, a validação e o teste utilizando as informações de todos os 32 sensores. Dessa forma, é possível verificar quais algoritmos conseguem prever os vazamentos de maneira satisfatória. Para tanto, analisou-se o desempenho geral das arquiteturas, bem como o comportamento individual por local de incidência de vazamento, conforme os processos descritos a seguir:

1. **Controle de Inicialização por (*Seed*):** Criou-se um vetor com 10 sementes pseudoaleatórias fixas. Essas sementes garantem a reprodutibilidade do experimento, permitindo avaliar o comportamento do modelo a partir de inicializações diferentes;
2. **Treinamento Multi-Semente:** O algoritmo foi treinado em 10 execuções independentes utilizando todos os sensores da rede. Em cada execução, uma das sementes previamente definidas foi utilizada para controlar os processos pseudoaleatórios do modelo, resultando em 10 versões treinadas da mesma arquitetura;
3. **Verificação do resultado global**: Por meio de um gráfico de boxplot, o desempenho dos modelos foi analisado de acordo com seu F1-score global;

4. **Verificação do resultado por classe**: Através de graficos de matriz confusão para cada um dos modelos avaliados buscou-se analisar quais pontos de vazamentos eram mais dificeis de serem identificados e quais as facilidade e dificuldades de cada um dos modelos.

4.7 EXPERIMENTO 2 - ANÁLISE SOB CONDIÇÃO DE FALHA SEQUENCIAL

Este experimento teve como objetivo analisar a confiabilidade e a estabilidade das arquiteturas de GNN sob condições de falha progressiva. A metodologia busca garantir que a capacidade do modelo de lidar com a perda de sensores seja consistente e não dependente de uma inicialização específica de seus pesos sinápticos. O procedimento para cada algoritmo é detalhado a seguir:

1. **Controle de Inicialização por (*Seed*):** Igual ao realizado no experimento 1;
2. **Treinamento Multi-Semente:** Igual ao realizado no experimento 1;
3. **Teste de Degradação:** Cada um dos modelos foi submetido a um teste de degradação que consiste na remoção progressiva de sensores, seguindo uma sequência de falhas idêntica para todas as rodadas. Este passo avalia como o desempenho decai conforme ocorrem falhas progressivas no número de sensores;
4. **Métricas:** Para cada um dos testes, calculou-se o desempenho mínimo, máximo, médio e o desvio padrão do F1-score entre as 10 execuções de cada modelo. O resultado é consolidado em um gráfico de linhas com barras de erro, que é composto por três elementos principais:
    - **Linha Central:** Conecta os pontos que representam a média de cada categoria;
    - **Pontos Centrais:** Localizado exatamente no valor médio;
    - **Barras de Erro:** Demonstra as variações que as sementes geraram devido à inicialização aleatória dos pesos.

Desta forma, buscou-se responder a duas perguntas cruciais: a) Qual a tendência do modelo quando há falha sequencial nos sensores e, se os resultados são consistentes, descartando-se algum viés relacionado a uma semente específica. Uma barra de erro curta sugere que o impacto da retirada deste sensor é real. Da mesma forma, uma barra muito longa sugere uma alta dependência do modelo em relação à inicialização dos pesos.

Para garantir que os resultados sejam consistentes na comparação entre os modelos manteve-se a mesma sequência de remoção dos sensores. Esta sequência foi elaborada de

forma pseudoaleatória para simular uma situação real em que os sensores podem falhar em lugares distintos. Está sequência é apresentada abaixo:

- 11, 10, 12, 13, 14, 15, 32, 28, 20, 16, 17, 24, 02, 29, 31, 22, 25, 06, 23, 30, 03, 26, 19, 21, 09, 04, 05, 27, 08 , 07, 18, 01.

4.8 EXPERIMENTO 3 - AVALIAÇÃO DA IMPORTÂNCIA DOS SENSORES NA SITUAÇÃO DE FALHA OPERACIONAL

Este experimento teve como objetivo mapear a relevância de cada nó para cada modelo analisado, identificando quais sensores fornecem as informações mais relevantes para a tomada de decisão de cada algoritmo. Para isolar o impacto de cada sensor e evitar o efeito cascata de múltiplas falhas, adotamos um método de remoção unitária controlada. O procedimento é detalhado a seguir:

1. **Controle de Inicialização por (*Seed*):** Igual ao realizado no experimento 1;
2. **Treinamento Multi-Semente:** Igual ao realizado no experimento 1;
3. **Avaliação e Remoção Unitária:** Para cada uma das 10 simulações por modelo treinado, estabeleceu-se um valor de F1-score de referência. Em seguida, cada um dos 32 sensores foi removido individualmente e de forma isolada, sempre retornando à configuração de rede completa antes da próxima remoção;
4. **Cálculo do Impacto ($\Delta F1$):** A importância de cada sensor i foi quantificada pela queda de desempenho provocada por sua ausência, calculada como:

   $$\Delta F1(i) = F1_{completo} - F1_{sem_sensor_i}$$

   Este processo resulta em 10 valores de impacto para cada sensor, permitindo o cálculo das demais métricas (média, mínimo, máximo e desvio padrão);
5. **Mapeamento :** Os resultados são consolidados em duas frentes:
   - **Variação do F1-score:** Um gráfico de barras com barras de erro que identifica visualmente os sensores com maior $\Delta F1$;
   - **Mapa de Calor da RDA:** Uma representação sobre a topologia real da rede de Hanói, onde a cor de cada nó (do azul ao vermelho) indica seu grau de importância para os resultados de cada um dos modelos avaliados.

Com o intuito de facilitar a análise dos dados, escolheram-se dois gráficos para representar os resultados:

- **Gráfico de Barras com Barra de Erro:** Representa a queda do desempenho do modelo quando algum sensor é removido. O $\Delta F1$ representa a queda do F1-score quando um determinado sensor é removido. Assim, quanto maior a barra, mais importante é o sensor para a rede. As barras de erro representam a variação dos resultados de acordo com a inicialização aleatória das sementes;
- **Mapa de calor sobre a RDA analisada:** busca demonstrar na própria RDA analisada quais são os sensores mais importantes para o modelo. A importância é demonstrada por um gradiente de cores em que os **nós Vermelhos** são pontos vitais da rede, onde sensores localizados nestas posições são de alta importância para o aprendizado do modelo. Por outro lado, **nós Azuis** são sensores que fornecem informações pouco relevantes.

A análise da importância individual dos sensores, medida através da queda de desempenho ($\Delta F1$), permite avaliar o grau de dependência que cada modelo desenvolve em relação a nós específicos da topologia.

4.9 EXPERIMENTO 4 - AVALIAÇÃO DOS SENSORES CRÍTICOS PARA O APRENDIZADO DO MODELO

O foco deste experimento foi identificar a importância de cada sensor para a capacidade de aprendizado dos modelos. O objetivo principal foi determinar quais nós da RDA fornecem as informações mais relevantes para que o modelo construa seu conhecimento, assim como, quais sensores sensores podem ser removidos sem uma diminuição significativa no desempenho.

- **Controle de Inicialização por (*Seed*)**: Da mesma forma que os experimentos anteriores;
- **Exclusão e Retreinamento**: Para cada um dos 32 sensores, o sistema realiza um novo ciclo completo: **a)** O sensor é removido do dataset de treino e de teste. **b)** O algoritmo é treinado do zero 10 vezes sem aquele sensor específico. **c)** Calcula-se um novo valor de F1-score;
- **Cálculo do Impacto ($\Delta F1$):** A importância de um sensor é definida pela queda no desempenho que sua ausência causa:

$$\Delta F1 = (\text{Média com 32 sensores}) - (\text{Média sem o sensor analisado})$$

- **Análise e Visualização dos Dados**: Os resultados foram analisados por meio de dois gráficos principais: **a) Gráfico de Barras** Exibe o $\Delta F1$ para cada sensor e

b) Mapa de Calor da RDA: Os resultados são projetados diretamente sobre o mapa da RDA, onde: Nós vermelhos são Sensores essenciais, sem os quais o modelo não consegue aprender, e nós azuis que são sensores redundantes, cuja ausência não influência o desempenho do modelo.

Os resultados foram exibidos na forma de gráficos de barra com o intuito de ilustrar o impacto no desempenho quando o modelo é treinado desde o início sem ter visto os dados de um sensor específico. Barras altas sugerem que aquele nó em questão possui informações relevantes para o aprendizado do modelo e sem essa informação o modelo não consegue apresentar um bom resultado combinando os dados dos demais sensores. O mapa de calor é similar ao apresentado no experimento 2 e representa os locais onde a instalação do sensor é obrigatória. Sem estes valores, o modelo é incapaz de aprender corretamente as variações da RDA.

A avaliação do impacto da exclusão de sensores específicos durante a fase de treino revela quais nós fornecem as informações fundamentais para a convergência dos modelos e quais atuam como elementos redundantes ou, em alguns casos, prejudiciais. A métrica $\Delta F1_{train}$ ilustra a diferença de desempenho entre o modelo treinado com a rede completa e o modelo treinado sem o sensor em análise.

4.10 EXPERIMENTO 5 - DETERMINAÇÃO DO NÚMERO MÍNIMO DE SENSORES PARA TREINAMENTO

O foco desse experimento foi mapear a relação entre a quantidade de sensores e a capacidade preditiva do sistema. Nele busca-se identificar a quantidade mínima de sensores necessária para manter um desempenho aceitável:

- **Amostragem combinatória de sensores (k)**: Este experimento utiliza uma abordagem baseada em subconjuntos de sensores para garantir que os resultados obtidos não sejam decorrentes de uma seleção particularmente favorável de sensores;
- **Aumento gradativo do número de sensores**: Definiu-se 11 quantidades de sensores (4, 6, 8, 10, 12, 15, 20, 25, 28, 30 e 32) para serem realizados treinamentos;
- **Múltiplas combinações (M)**: Para cada quantidade k, não testamos apenas um grupo de sensores, mas realizamos o sorteio de 5 combinações diferentes. Essa estratégia é fundamental para reduzir vieses associados a configurações específicas de sensores, uma vez que subconjuntos menores e melhor posicionados podem apresentar desempenho superior a conjuntos maiores distribuídos de forma menos eficiente;

- **Treinamento com seeds:** Para cada combinação de sensores, o modelo é treinado 5 vezes do zero. Com isso é possível analisar tanto a incerteza da inicialização aleatória dos algorítmos quanto a localização dos sensores pertencentes aos subconjuntos.

Para exibição dos resultados foi gerada uma curva relacionando o F1-score com o número de sensores da rede, acompanhada das respectivas barras de erro. Pontos de inflexão nas curvas tipicamente indica onde o modelo inicia a aprendizagem e o ponto onde o desempenho estabiliza. A curva central, que é a média dos valores de F1-score, representa o desempenho esperado se apenas um número pseudoaleatório de sensores for treinado. A avaliação dos modelos em função da expansão progressiva do número de sensores da rede demonstra um número mínimo necessário para o modelo aprender, além de evidenciar que o desempenho da rede não depende apenas do volume de dados, mas está altamente relacionado à localização dos nós selecionados.

5 RESULTADOS E DISCUSSÕES

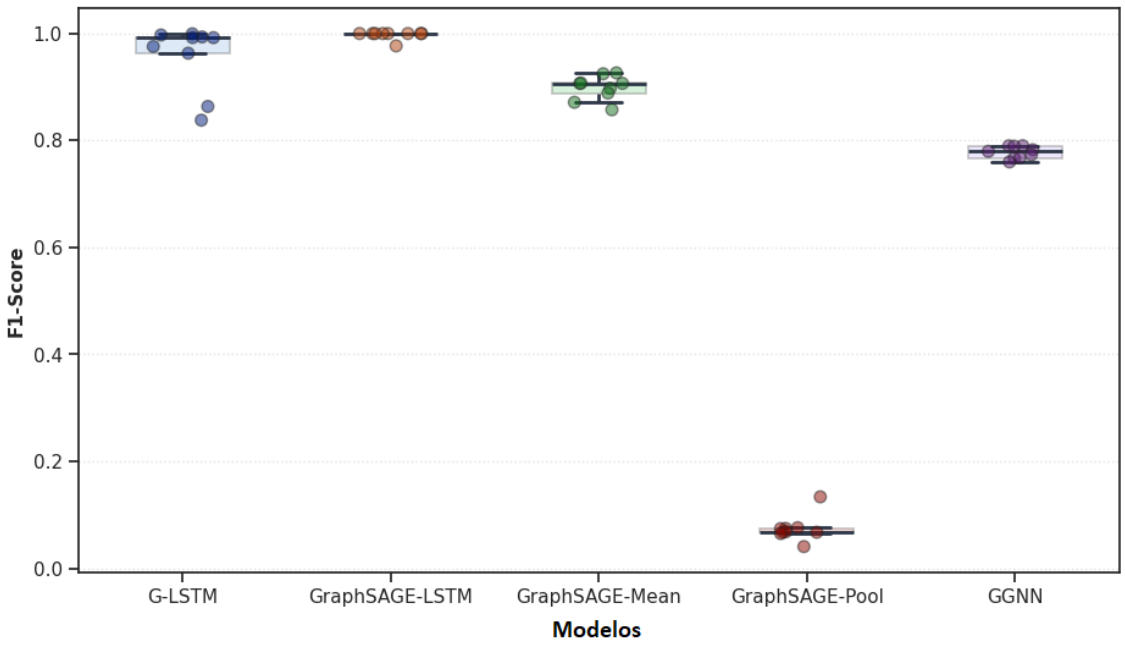
Este capítulo tem como objetivo apresentar de maneira detalhada os resultados obtidos com os experimentos realizados durante o desenvolvimento desta pesquisa.

5.1 EXPERIMENTO 1 - ANÁLISE SOB CONDIÇÃO IDEAL

5.1.1 ANÁLISE GLOBAL DOS RESULTADOS

A Figura 14 apresenta o F1-score obtido por cada um dos cinco modelos avaliados quando treinados e testados sob condições ideais, ou seja, com a aquisição de dados de todos os 32 sensores da RDA de Hanói. Esta análise permitiu compreender o limite máximo de aprendizado de cada modelo isoladamente, considerando as 10 diferentes inicializações aleatórias (sementes). Observou-se que os modelos que utilizavam LSTM obtiveram os melhores resultados. O modelo GraphSAGE-LSTM apresentou o melhor resultado global, alcançando um F1-score mediano praticamente perfeito (próximo a 1,00) e com baixíssima dispersão.

Figura 14 – Comparativo do F1-score dos modelos treinados com os dados de todos os sensores.



O G-LSTM também demonstrou excelente capacidade de detecção e localização, apresentando apenas uma leve variação (dispersão) entre as inicializações e dois resultados considerados como *outliers*.

O modelo GraphSAGE-Média obteve um desempenho satisfatório e consistente, atingindo um F1-score mediano em torno de 0,90. O modelo GGNN, por sua vez, apresentou um F1-score mediano abaixo dos 80%, o menor entre as 3 arquiteturas analisadas anteriormente. Por fim, o modelo GraphSAGE-Pooling destacou-se negativamente com um valor abaixo de 0,10 em todas as execuções. Assim, inferiu-se que este modelo realizou previsões puramente aleatórias.

Desta forma, sob condições ideais de monitoramento, as variações do GraphSAGE (com exceção do pooling) e o G-LSTM destacaram-se pela precisão máxima. O GGNN obteve bons resultados, mas ainda inferiores aos modelos citados acima.

5.1.2 INFLUÊNCIA DA LOCALIZAÇÃO DOS VAZAMENTOS

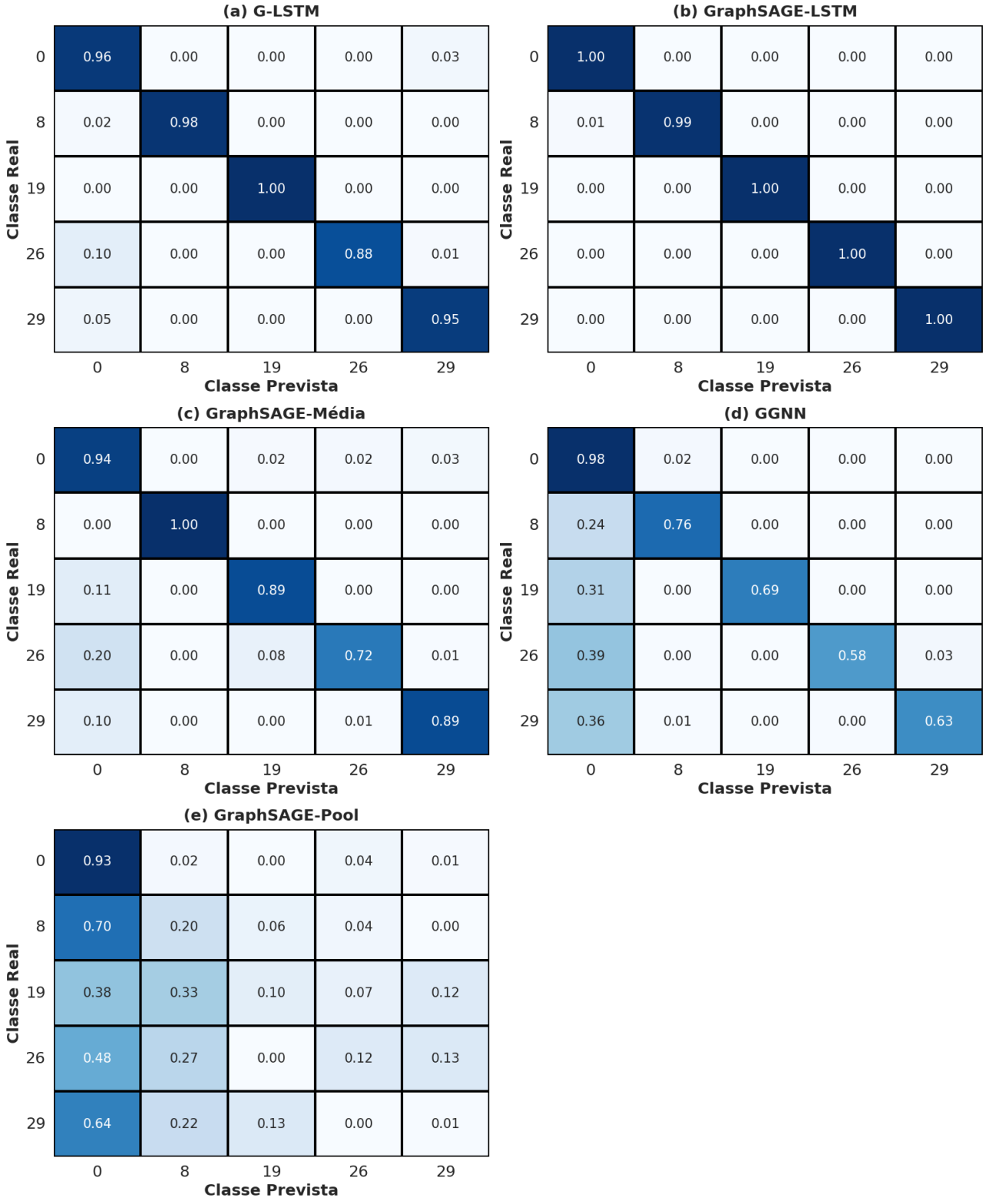
Conforme estabelecido no procedimento experimental, além do cenário de operação normal (classe 0), os vazamentos foram simulados em tubulações espalhadas pela RDA de Hanói e classificados de acordo com a proximidade aos nós 8, 19, 26 e 29. O objetivo desta análise foi verificar se a localização do vazamento exercia influência na capacidade preditiva das redes neurais avaliadas.

A Figura 15 apresenta as matrizes de confusão dos modelos utilizados. Os resultados evidenciam que os modelos que utilizavam LSTM em sua arquitetura obtiveram uma maior eficiência, alcançando resultados muito próximos do ideal. Entretanto, ressalta-se que isso é um risco, pois estes modelos podem se ajustar demais aos dados de treinamento e apresentar *overfitting*.

O GraphSAGE com agregador do tipo média também alcançou ótimos resultados. Entretanto, este modelo apresentou uma maior dificuldade em prever a incidência de vazamento próximo ao nó 26, sendo a única classe deste modelo com o número de acertos próximo dos 70%. O GGNN também apresentou um bom desempenho. Contudo, este modelo foi tendencioso em classificar a saída como zero (sem incidência de vazamento). Com isso, todas as demais classes tiveram dados classificados erroneamente como zero.

Por fim, conforme esperado, o modelo de GraphSAGE que utiliza o *pooling* como agregador falhou em classificar todas as classes, justificando os resultados encontrados no boxplot da Figura 14. Notou-se que, na maior parte dos casos, este modelo aprendeu a prever a classe 0 ao invés da classe correta. Isso provavelmente ocorreu devido ao desbalanceamento da quantidade de classes, em que a classe 0 tem o dobro do número de simulações das demais, de forma que o modelo aprendeu a ”chutar” a classe com maior número de exemplares.

Figura 15 – Matriz de confusão normalizada para os modelos avaliados.



Fonte: Elaborada pelo autor.

As próximas seções buscaram analisar os modelos testados em cenários operacionais adversos, onde ocorre a perda de sensores, assim como, identificar o número mínimo de

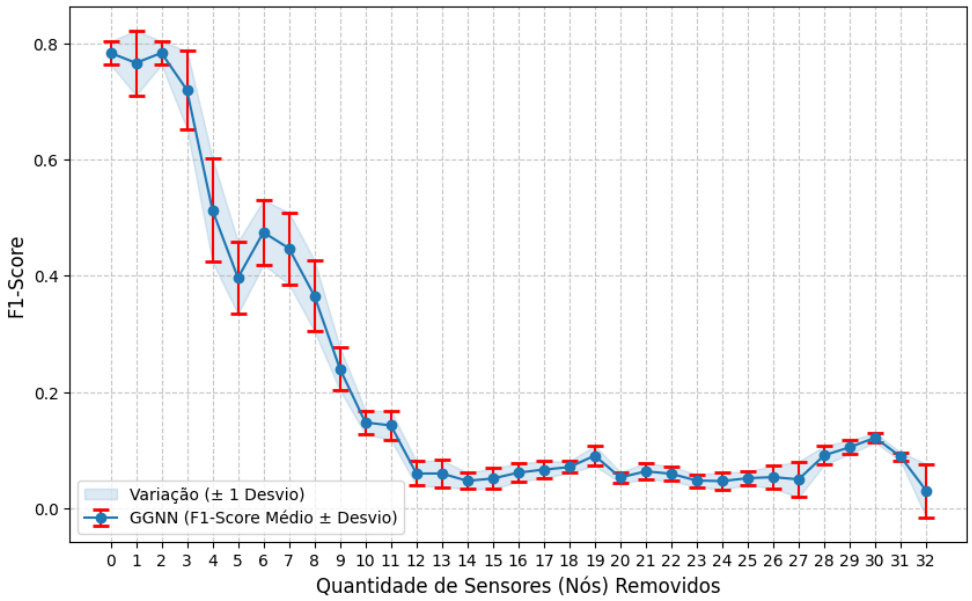
sensores necessários para o modelo aprender. Apesar do modelo GraphSAGE-Pooling não apresentar bons resultados, optou-se por realizar o experimento 2 também neste modelo para verificar se há alguma melhora no desempenho com a retirada progressiva dos sensores.

5.2 EXPERIMENTO 2 - ANÁLISE SOB CONDIÇÃO DE FALHA SEQUENCIAL

A avaliação da resiliência dos modelos face à remoção progressiva de nós (sensores) revelou comportamentos distintos entre as arquiteturas de GNN testadas. A análise do F1-score, acompanhada do desvio padrão obtido através das 10 inicializações aleatórias (*seeds*), nos permitiu tirar conclusões fundamentais sobre a robustez e a dependência topológica de cada abordagem. Os resultados deste experimento estão apresentados nas figuras 16 a 20.

5.2.1 GGNN

Figura 16 – Degradação por Falha dos Sensores para o Modelo GGNN.

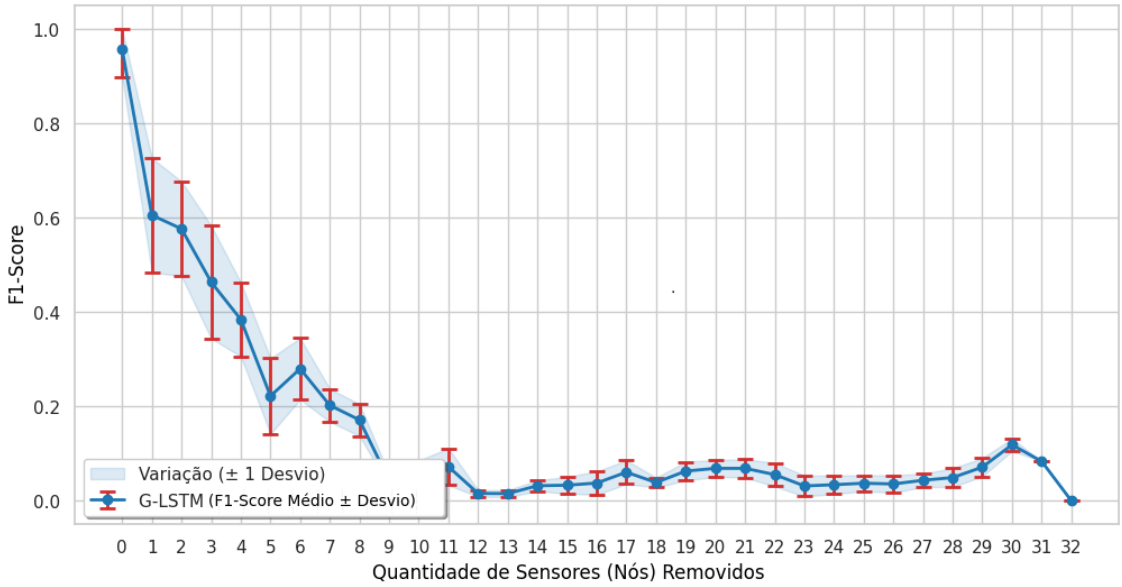


Apesar de a GGNN não ter alcançado o melhor resultado do treinamento com todos os valores dos sensores, ela demonstrou ser a mais equilibrada entre as avaliadas. O gráfico apresentou barras de erro pequenas ao longo de quase todo o processo de degradação, o que indicou um desempenho robusto às inicializações. Adicionalmente, embora como esperado ocorresse uma perda de desempenho com a redução do número de sensores, o

modelo exibiu uma queda suave e progressiva, e o F1-score não sofreu uma quebra abrupta com a remoção dos primeiros sensores.

5.2.2 G-LSTM

Figura 17 – Degradação por Falha dos Sensores para o Modelo G-LSTM.



O modelo G-LSTM apresentou o F1-score inicial próximo do ideal quando a rede estava completa. No entanto, ao retirar os dois primeiros sensores, observou-se um desvio padrão alto. Este comportamento sugeriu que, ao perder essas informações, o modelo se tornou instável e dependente das inicializações aleatórias. Após a remoção de 4 nós, a queda de desempenho acentuou-se e o modelo começou a estabilizar em valores residuais mais cedo do que a GGNN.

5.2.3 GraphSAGE-LSTM

Os resultados obtidos para a arquitetura GraphSAGE-LSTM indicaram a ocorrência de uma falha catastrófica global, caracterizada por elevada sensibilidade à remoção de sensores da rede. Apesar da baixa variabilidade associada às diferentes inicializações do modelo, a exclusão de apenas um único sensor provocou uma degradação de aproximadamente 80% no desempenho. Tal comportamento sugeriu forte dependência da informação proveniente de todos os sensores para a realização das predições, evidenciando limitada robustez estrutural frente à perda de nós de entrada.

Figura 18 – Degradação por Falha dos Sensores para o Modelo GraphSAGE-LSTM.

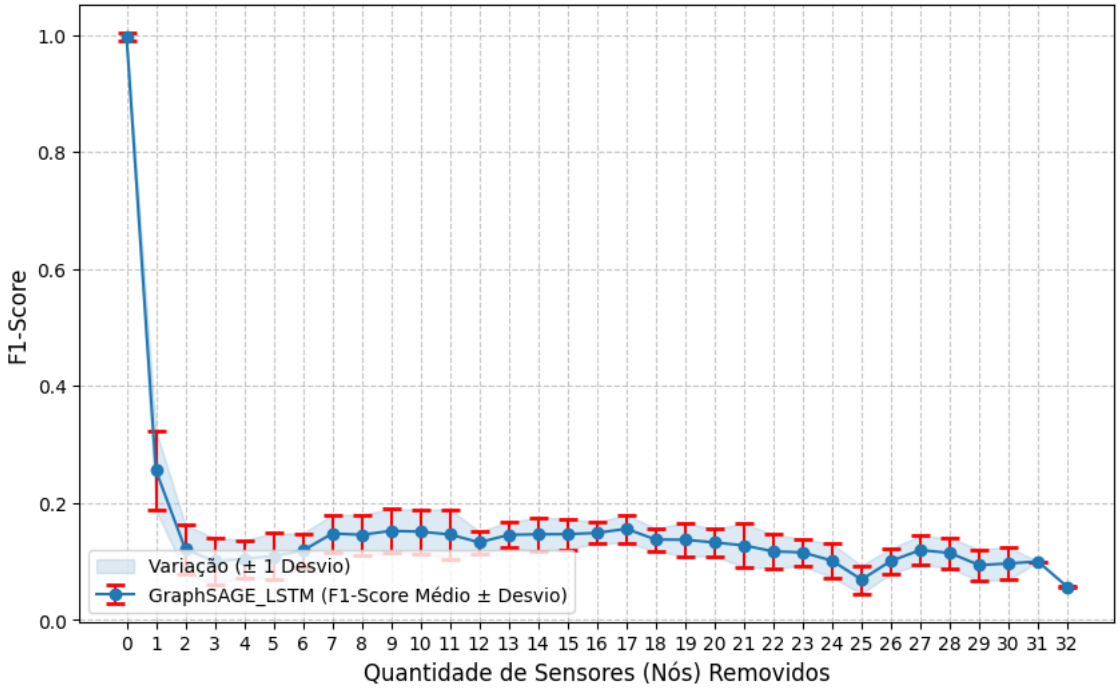
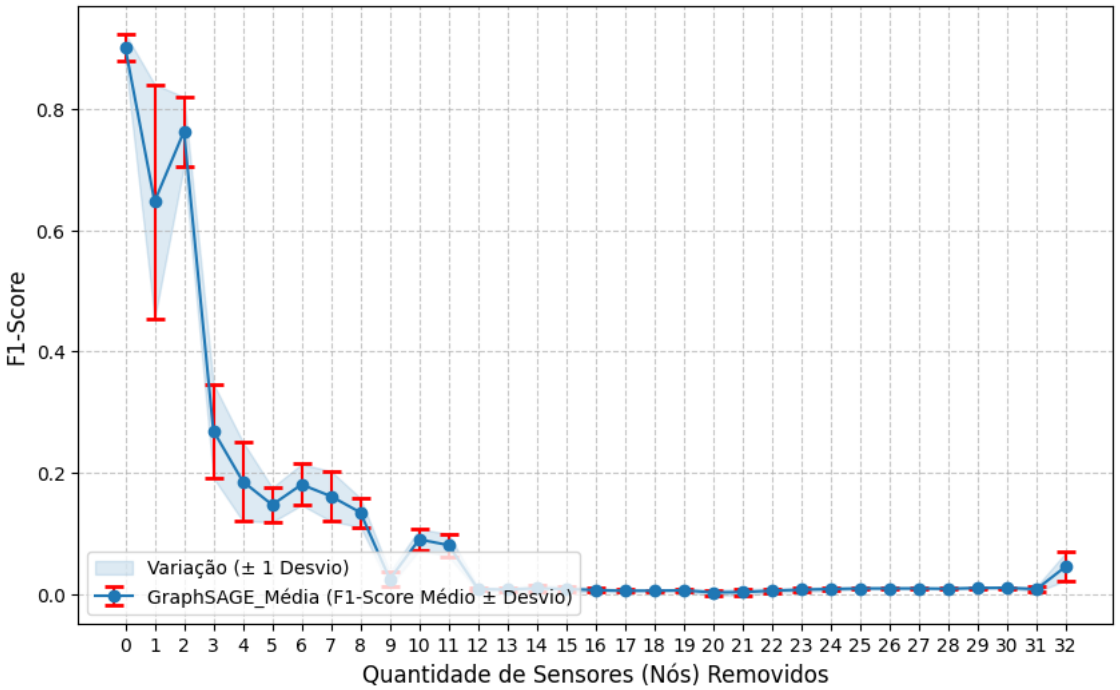


Figura 19 – Degradação por Falha dos Sensores para o GraphSAGE-Média.



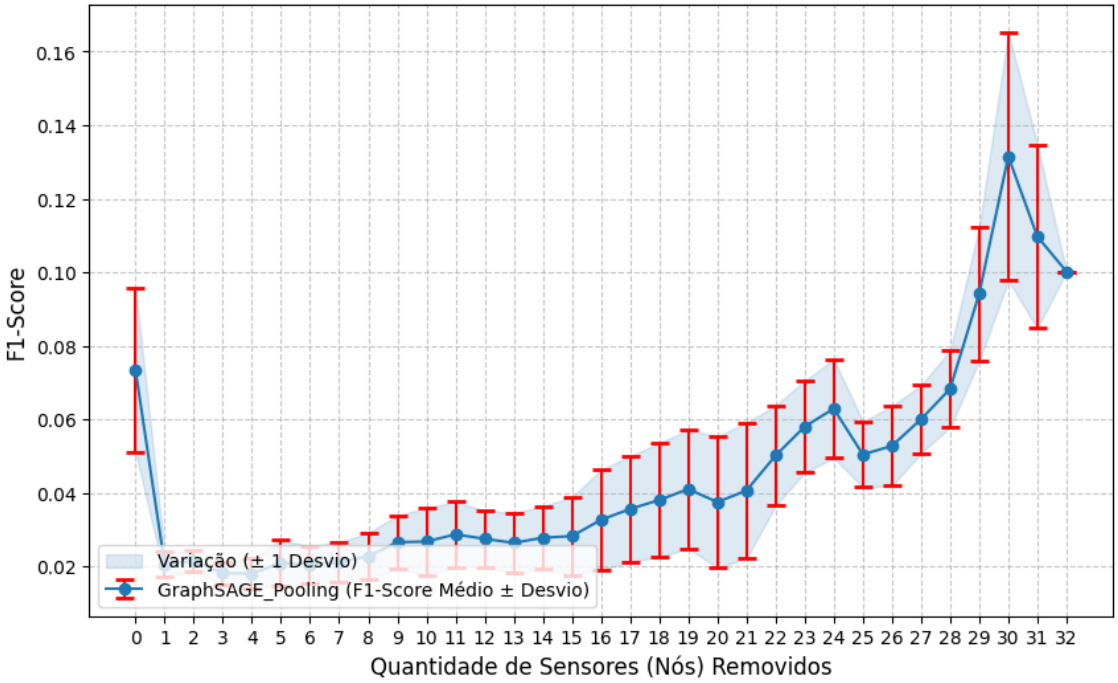
5.2.4 GraphSAGE-Média

Dentre as arquiteturas testadas, esta apresentou a estratégia de agregação mais simples, baseada na média dos atributos dos nós vizinhos durante o processo de propagação

de informações. A remoção de apenas um sensor resultou em elevada dispersão dos resultados, evidenciada pelo aumento expressivo do desvio padrão entre execuções. Com a exclusão de três sensores, o modelo passou a operar com F1-score próximo de 20%. Adicionalmente, esta foi a única arquitetura cuja degradação de desempenho ocorreu de maneira consistente até valores próximos de zero, indicando colapso progressivo da capacidade de generalização frente à perda de sensores da rede.

5.2.5 GraphSAGE-Pooling

Figura 20 – Degradação por Falha dos Sensores para o Modelo GraphSAGE-Pooling.



Em conformidade com o apresentado no estudo de (RODRIGUES; ROLLE; GODOY, 2024), este modelo não foi capaz de aprender quando havia ou não incidência de vazamentos. O F1-score inicial encontrava-se abaixo de 0,1, um valor indicativo de um desempenho não superior a uma classificação aleatória. O aumento observado no F1-score durante o processo de remoção dos sensores não demonstrou um aprendizado, e sim ruídos no processo.

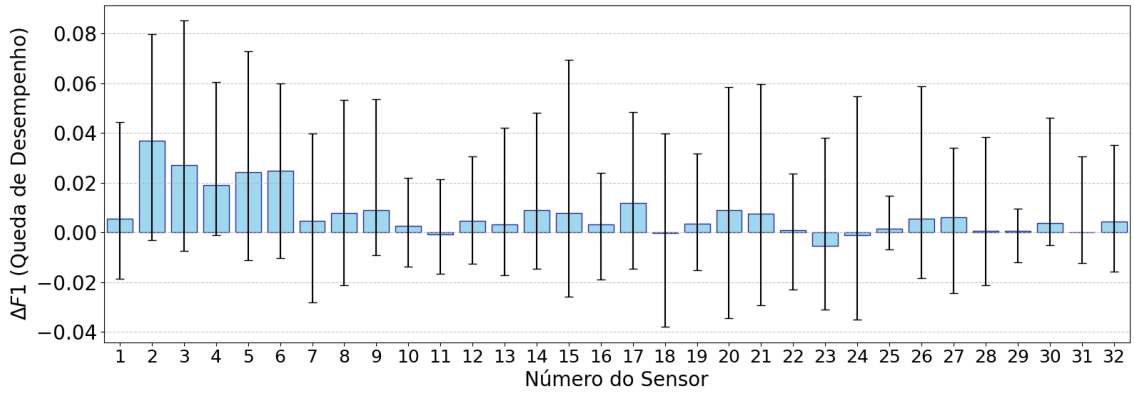
Por não apresentar nenhum aprendizado, o GraphSAGE com agregador do tipo *pooling* não foi mais considerado nos experimentos subsequentes.

5.3 EXPERIMENTO 3 - AVALIAÇÃO DA IMPORTÂNCIA DOS SENSORES NA SITUAÇÃO DE FALHA OPERACIONAL

As figuras 21, 22, 23 e 24, apresentadas nas seções a seguir, exibem os resultados obtidos para esse experimento.

5.3.1 GGNN

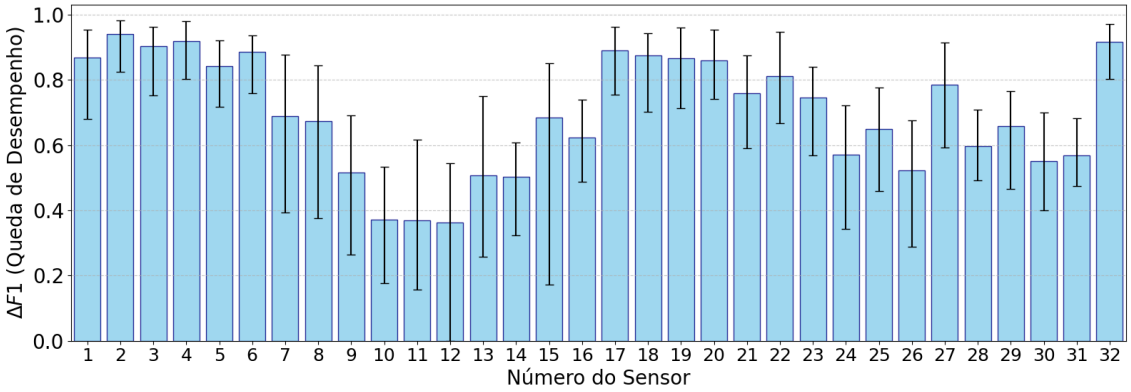
Figura 21 – Importância dos sensores para o Modelo GGNN.



O modelo GGNN apresentou um comportamento significativamente mais resiliente com relação à perda de sensores. Ao analisar o gráfico de barras, notou-se que a remoção unitária de qualquer um dos 32 sensores resultou em um valor $\Delta F1$ inferior a 0,1. Este fato demonstrou que a arquitetura não desenvolveu dependências críticas de nós específicos. A pequena magnitude do impacto indicou que o mecanismo de portas (*gating*) do GGNN foi altamente eficaz em extrair redundância da rede. A informação contida num sensor ausente foi provável e facilmente compensada através das mensagens propagadas pelos seus vizinhos, garantindo a estabilidade da rede neural.

5.3.2 G-LSTM

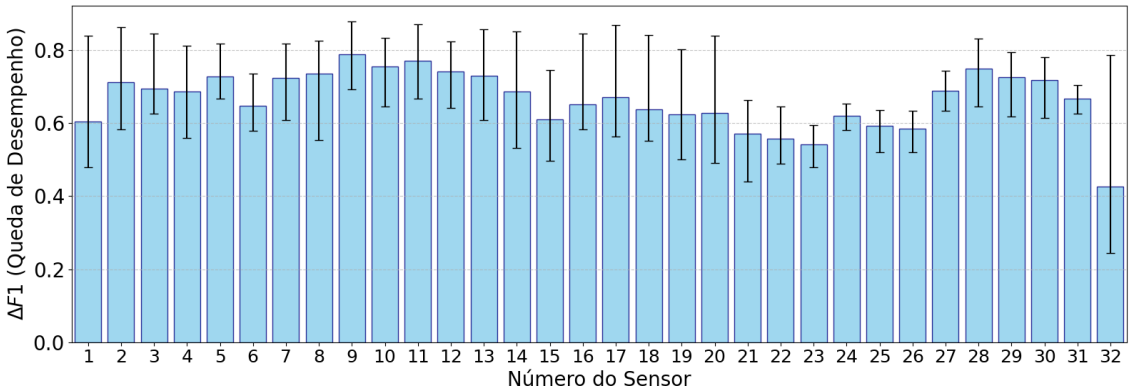
Figura 22 – Importância dos sensores para o Modelo G-LSTM.



Em contraste direto com o GGNN, esta arquitetura revelou uma alta vulnerabilidade à perda de nós individuais. Observou-se que a remoção de qualquer sensor da rede provocou uma queda drástica no desempenho. Esta queda elevada corroborou a hipótese de que o modelo precisava de todos os nós da rede para realizar suas predições. Adicionalmente, as barras de erro longas evidenciam uma alta dependência da inicialização dos pesos controlada pelas sementes.

5.3.3 GraphSAGE-LSTM

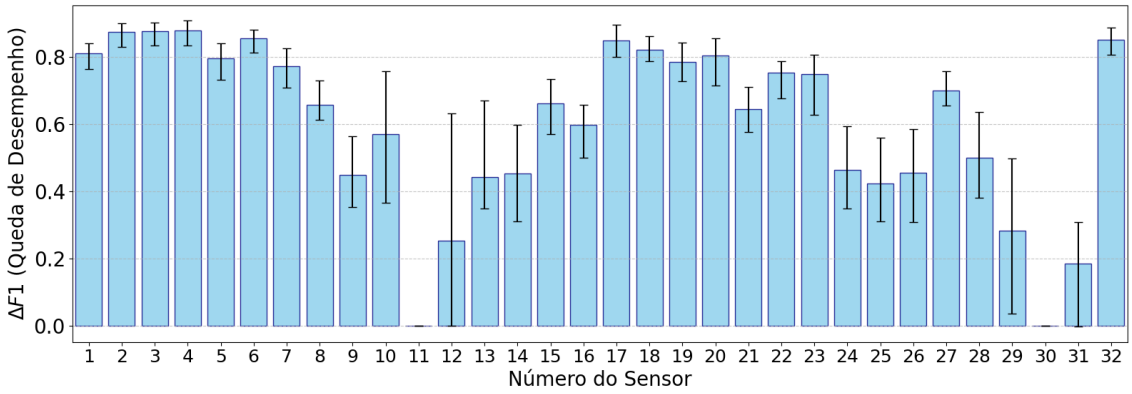
Figura 23 – Importância dos sensores para o Modelo GraphSAGE-LSTM.



Similar ao apresentado pelo G-LSTM, esta arquitetura mostrou-se altamente dependente dos dados de todos os sensores, o que fundamenta a hipótese de que os modelos de GNN que utilizam LSTM em sua arquitetura possuem esta característica.

5.3.4 GraphSAGE-Média

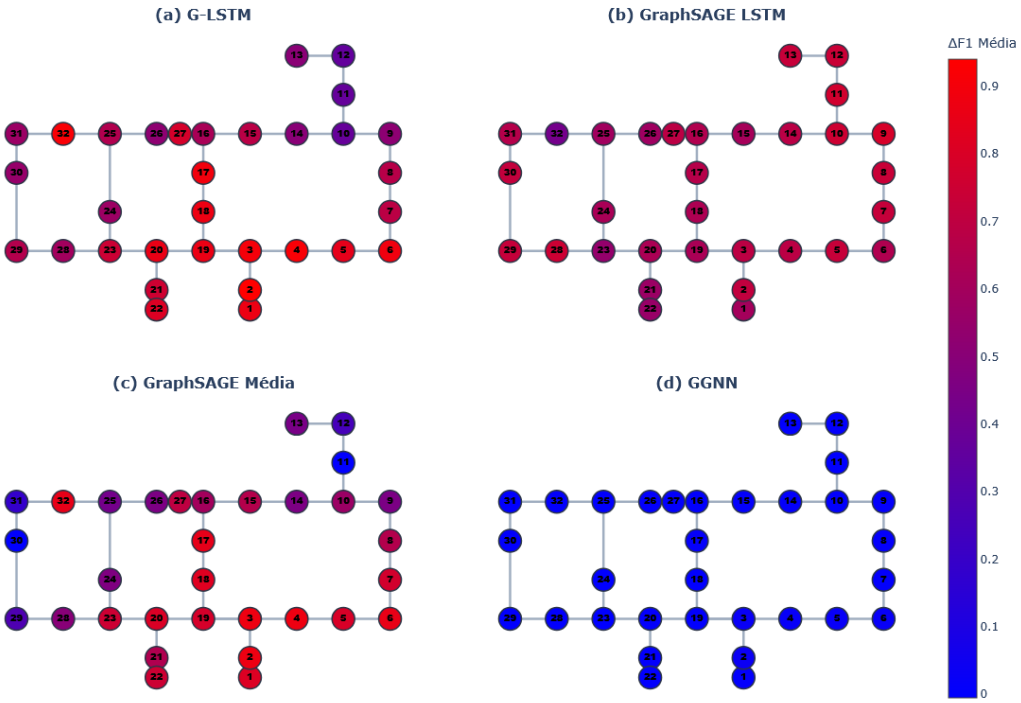
Figura 24 – Importância dos sensores para o Modelo GraphSAGE-Média.



Este modelo apresentou uma dependência assimétrica. Embora a maioria dos nós tenha provocado um alto impacto quando removidos (similar ao G-LSTM e ao GraphSAGE-LSTM), existiram exceções notáveis cuja ausência não afetou o modelo ($\Delta F1$ próximo de zero). No entanto, alguns sensores apresentaram barras de erro muito grandes, o que indicou que este modelo foi dependente das inicializações aleatórias quando alguns sensores específicos falharam.

5.3.5 MAPA DE CALOR

Figura 25 – Mapa de Calor da Importância dos sensores.



A projeção dos valores de queda de desempenho $\Delta F1$ sobre a topologia da RDA de Hanói por meio do mapa de calor (Figura 25) permitiu uma visualização da distribuição topológica da importância de cada sensor perante falhas operacionais.

Ao analisar o gráfico, notou-se que o modelo GGNN destacou-se por apresentar uma rede predominantemente composta por tons azuis, evidenciando sua superioridade, quando comparado aos demais modelos, na capacidade de extrair informações da rede mesmo durante a falha de um dos sensores.

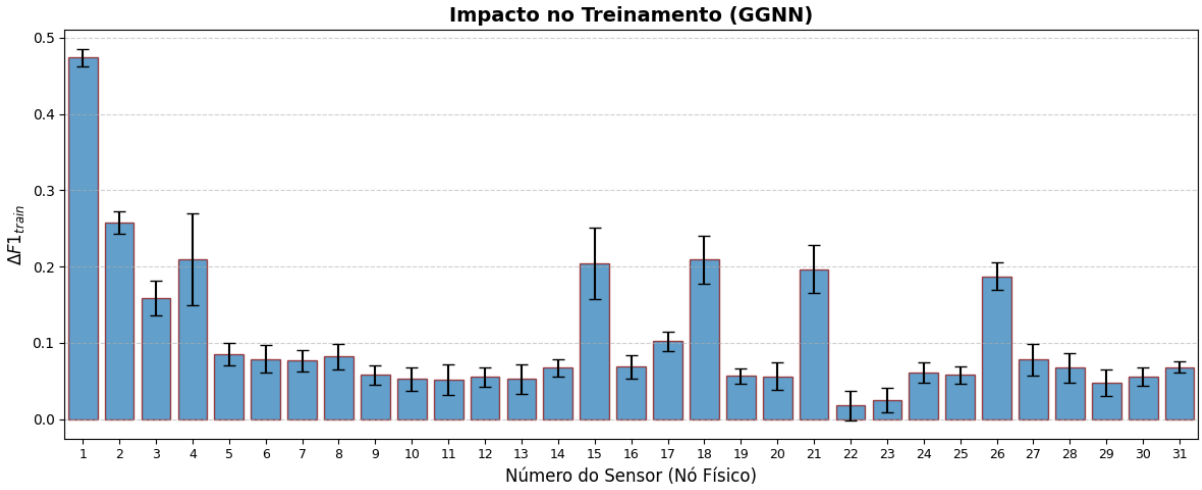
Em contrapartida, as arquiteturas G-LSTM e GraphSAGE-LSTM apresentaram mapas massivamente vermelhos. Isso ilustra a fragilidade destes modelos perante a perda de apenas um sensor, ou seja, estas redes mostraram-se altamente dependentes do funcionamento simultâneo de praticamente todos os nós da RDA. O modelo GraphSAGE-Média, por sua vez, também apresentou uma alta fragilidade; contudo, exibiu um comportamento assimétrico, com a predominância de tons vermelhos.

5.4 EXPERIMENTO 4 - AVALIAÇÃO DOS SENSORES CRÍTICOS PARA O APRENDIZADO DO MODELO

As figuras 26, 27, 28 e 29, apresentadas nas seções a seguir, exibem os resultados obtidos para esse experimento.

5.4.1 GGNN

Figura 26 – Sensores críticos para o modelo GGNN.



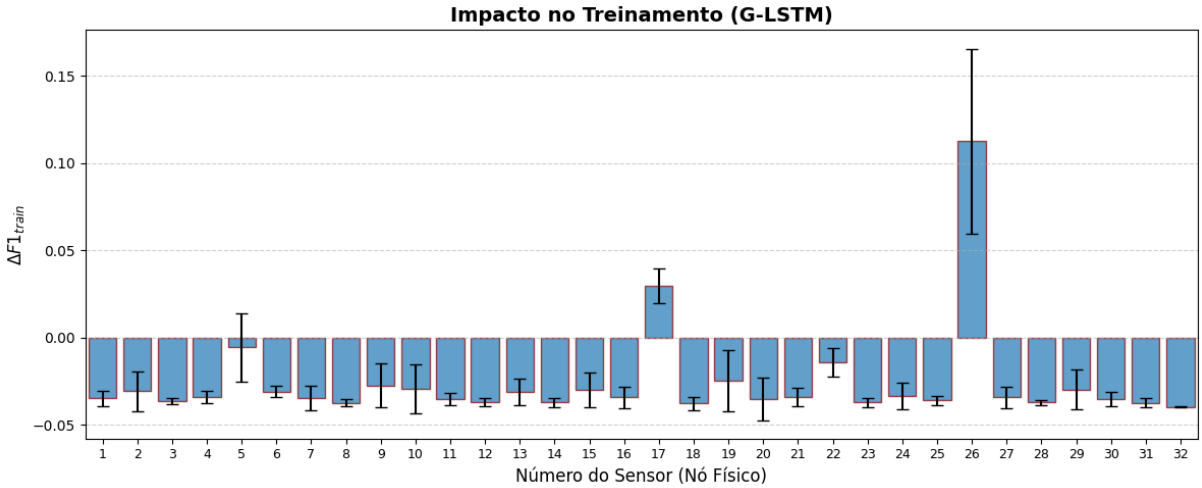
Ao contrário do Experimento 2, onde a arquitetura GGNN apresentou redundância quase total, a fase de treino revelou que o aprendizado dependia da maioria dos nós.

Observou-se uma alta dependência estrutural do sensor localizado no nó 1, com $\Delta F1_{train}$ próximo dos 0,5. Isso é explicável pelo fato de o nó 1 representar a entrada do sistema (logo após o reservatório principal), por onde flui toda a água consumida pela rede. Embora a maioria dos sensores apresentasse um impacto baixo, alguns sensores (ex: 2, 4, 15, 17, 21 e 26) também foram necessários para que o modelo treinasse satisfatoriamente. É importante observar que, dos sensores com informações importantes para o modelo aprender, apenas o sensor de número 26 encontra-se próximo de um dos locais de vazamento, sendo inclusive um dos rótulos deste experimento. Isso evidenciou que, na maioria dos casos, a importância de cada sensor foi definida de acordo com o modelo analisado, e não necessariamente com a proximidade do sensor em relação ao local onde ocorreu o vazamento.

Este resultado corroborou o comportamento exibido pelo modelo no Experimento 2. Embora praticamente todos os sensores fossem necessários para a etapa de aprendizado, quando ocorria uma falha isolada (como visto no Experimento 2), a arquitetura conseguiu superar a ausência dessa informação por meio da análise conjunta dos dados provenientes dos demais sensores.

5.4.2 G-LSTM

Figura 27 – Sensores Críticos para o modelo G-LSTM.

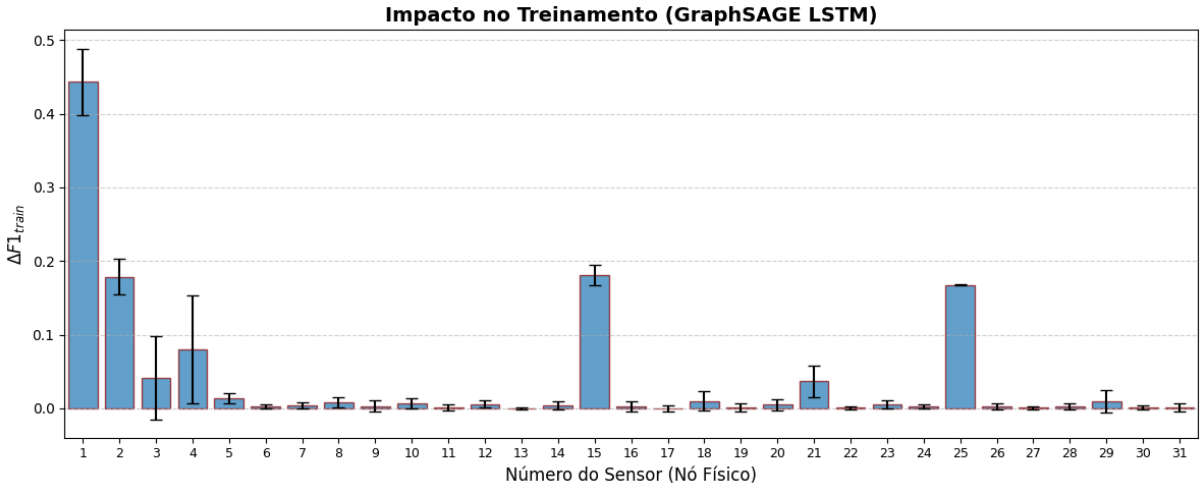


Os resultados evidenciaram uma dinâmica peculiar. Constatou-se que a exclusão da grande maioria dos sensores resultou num $\Delta F1_{train}$ negativo, ou seja, uma melhora no modelo quando foi treinado sem esses nós. Esse resultado sugeriu uma dificuldade de lidar com dados redundantes ou ruído quando expostos a todos os dados de treinamento. A exceção esteve nos sensores 26 e 17, cujas ausências provocaram uma queda no F1-score. O sensor 26 apresentou a maior barra de erro deste gráfico, o que indicou uma instabilidade do modelo quando este sensor não estava presente no treinamento. Similar ao apresentado

pelo GGNN, o sensor 26 foi o único sensor localizado próximo de um vazamento que afetou negativamente o desempenho do modelo.

5.4.3 GraphSAGE-LSTM

Figura 28 – Sensores Críticos para o modelo GraphSAGE-LSTM.



Notou-se que este modelo precisou da informação de poucos nós para aprender. A arquitetura dependeu primeiramente do sensor localizado no nó 1 e de alguns outros, como o 2, 15 e 25. A grande maioria dos sensores restantes registrou um $\Delta F1_{train}$ próximo de zero, indicando que, durante o treino, a rede não extraiu informações relevantes desses sensores. Notou-se a ausência de dependência dos sensores próximos aos vazamentos.

5.4.4 GraphSAGE-Média

A utilização do GraphSAGE com agregador por média assemelhou-se à variante LSTM devido à forte dependência do nó 1. Contudo, o comportamento quanto à remoção dos demais sensores foi similar ao padrão observado no modelo G-LSTM, em que a exclusão da maior parte dos sensores permitiu que o modelo obtivesse melhores resultados. Assim como os demais, este modelo não apresentou dependência dos sensores próximos aos locais com incidência de vazamento.

5.4.5 MAPA DE CALOR

A projeção dos valores de $\Delta F1_{train}$ sobre a topologia da RDA de Hanói permitiu visualizar a distribuição espacial dos nós mais críticos para o aprendizado do modelo. Com exceção do G-LSTM, existiu uma dependência das redes em relação ao sensor instalado no nó 1. Topologicamente, isso justifica-se por este ser o nó de entrada do sistema. Ao

Figura 29 – Sensores Críticos para o modelo GraphSAGE-Média.

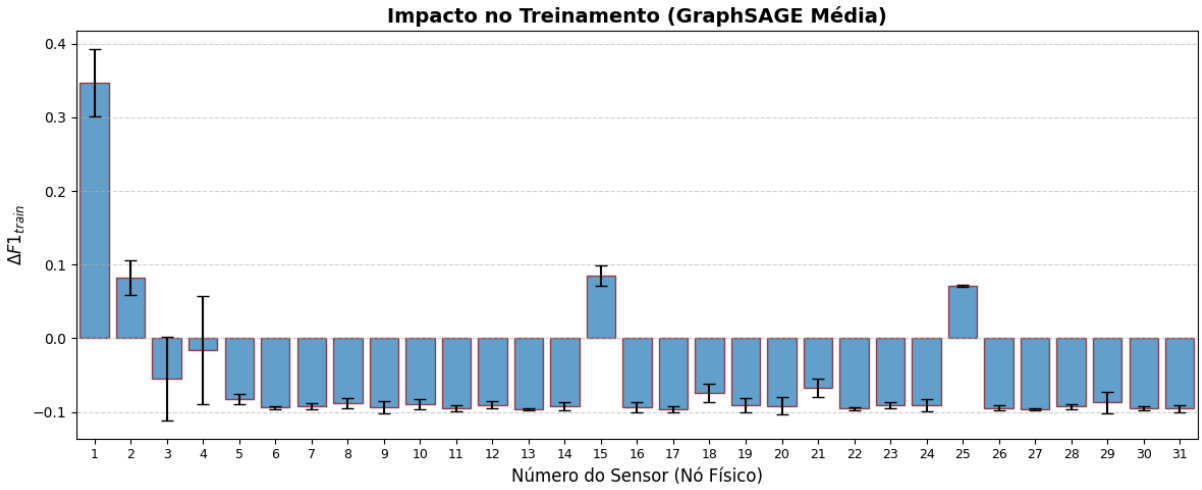
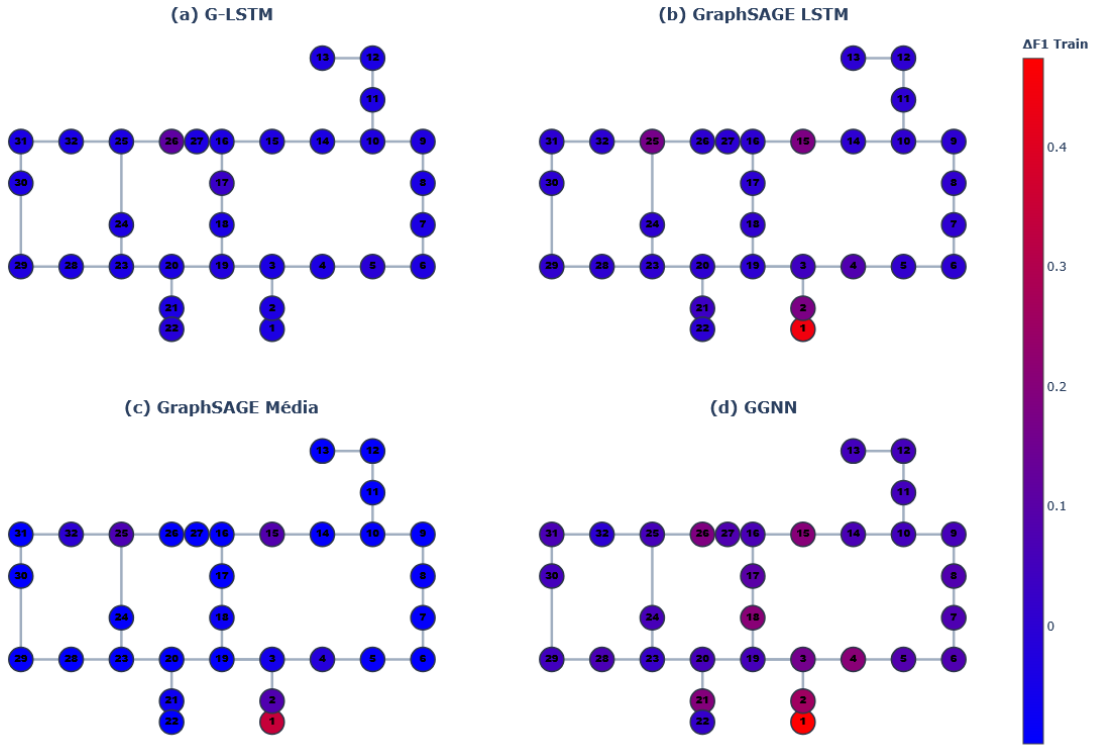


Figura 30 – Mapa de Calor.



removê-lo, não fornecendo dados deste para o treinamento, o modelo não conseguiu aprender.

A dependência do modelo G-LSTM em relação ao sensor localizado no nó 26 mostrou que para esta arquitetura este sensor possuía informações relevantes para o aprendizado do modelo. Ao comparar este resultado com a matriz de confusão demonstrada no experimento 1, notou-se que o sensor 26 foi o que apresentou menor desempenho de predição, o que

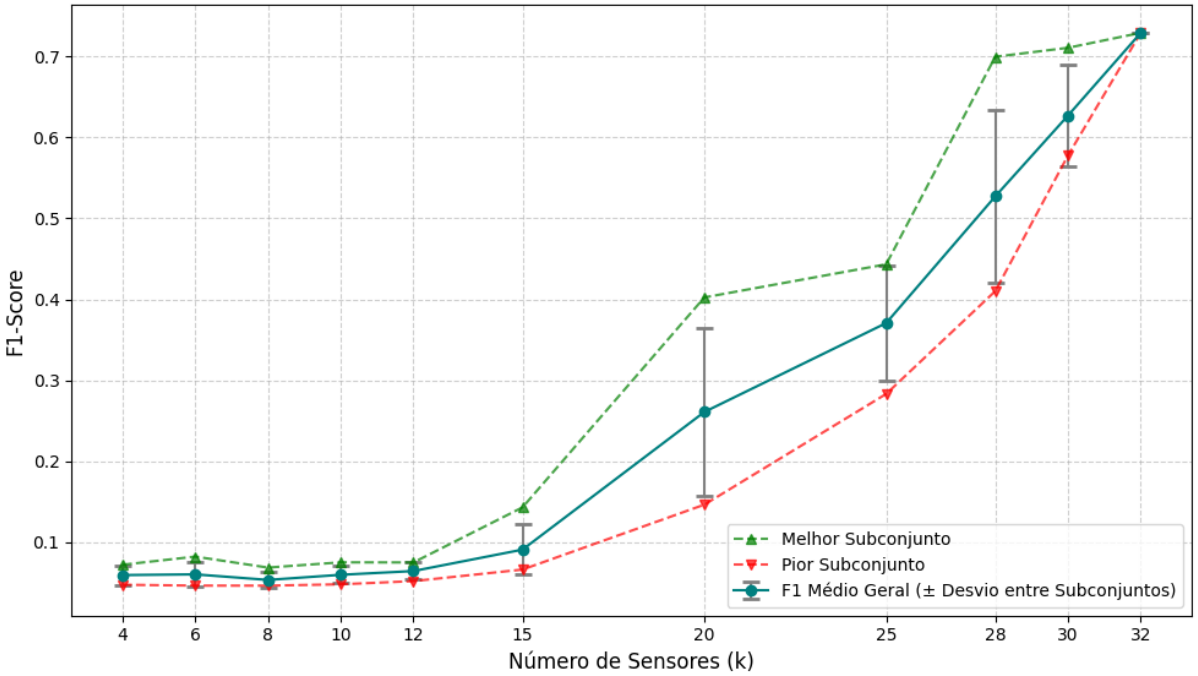
confirmou a importância desse sensor para a rede analisada.

5.5 EXPERIMENTO 5 - DETERMINAÇÃO DO NÚMERO MÍNIMO DE SENSORES PARA TREINAMENTO

As figuras 31, 32, 33 e 34, dispostas nas seções a seguir, trazem os resultados obtidos para este experimento.

5.5.1 GGNN

Figura 31 – Determinação do número mínimo de sensores para o GGNN.



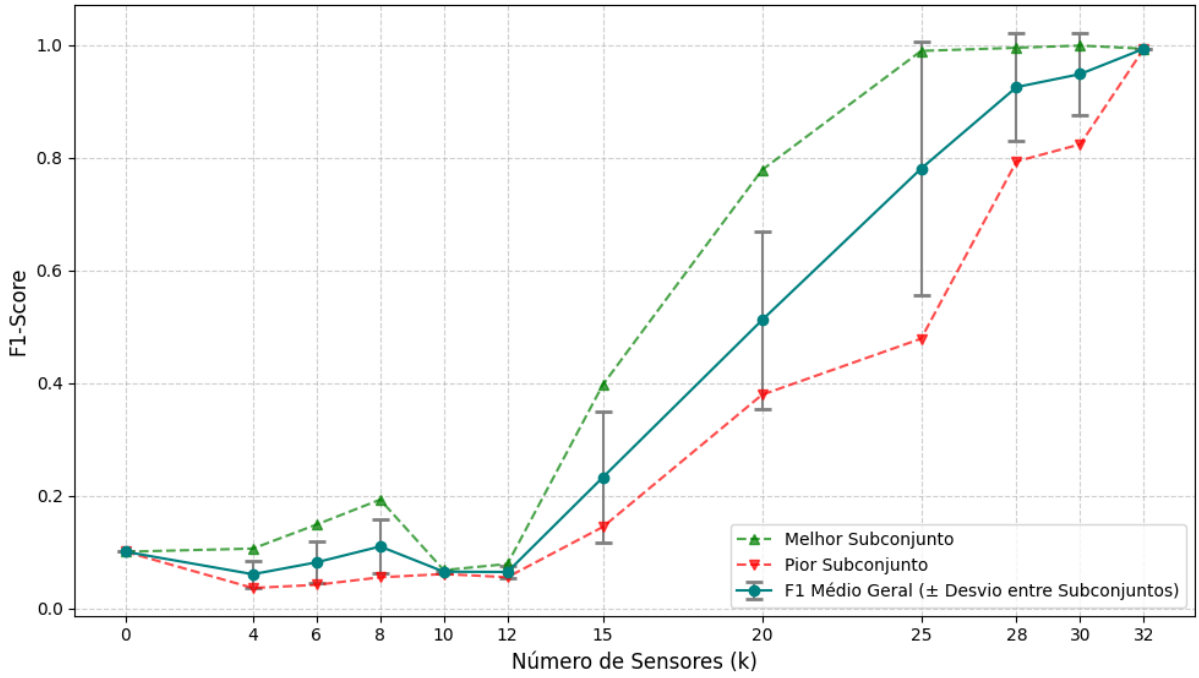
Fonte: Elaborada pelo autor.

Observou-se uma fase inicial prolongada em que o modelo começou a aprender apenas após 15 sensores. Após 20 sensores, o GGNN iniciou uma rampa de aprendizagem consistente. Notou-se também uma diferença significativa entre o desempenho do melhor e do pior subconjunto de sensores. Isso evidencia que, em um cenário com escassez de dados, a escolha dos sensores foi determinante para o desempenho do modelo.

5.5.2 G-LSTM

Os primeiros ganhos significativos no desempenho do modelo começaram a aparecer com os dados de 15 sensores. Um ponto de destaque foi que o modelo, na sua melhor seleção de subconjunto, conseguiu atingir resultados excelentes, próximos do ideal, com

Figura 32 – Determinação do número mínimo de sensores para o G-LSTM.



Fonte: Elaborada pelo autor.

apenas 25 nós. Ao analisar o comprimento das barras de erro durante a rampa do processo de aprendizado, notou-se uma dependência clara da localização topológica dos sensores durante o processo de aprendizado.

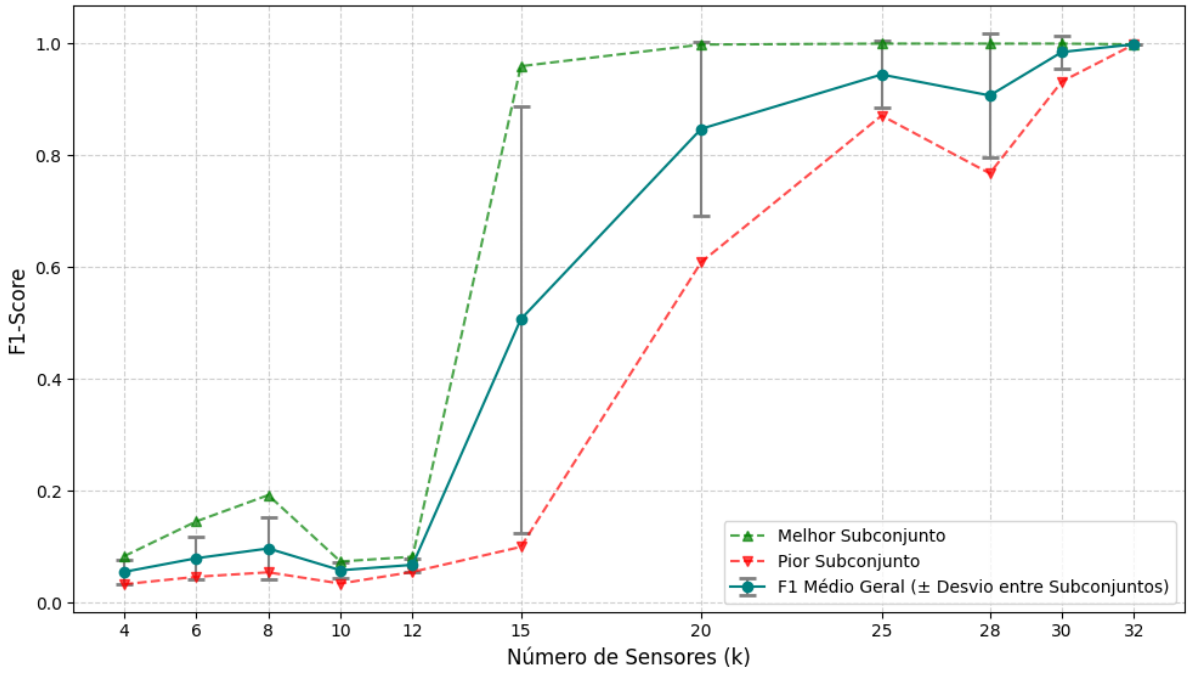
5.5.3 GraphSAGE-LSTM

O modelo apresentou a dependência mais severa em relação ao subconjunto selecionado. Já com 15 sensores, o modelo foi capaz de chegar a valores próximos do ideal quando o melhor subconjunto foi selecionado. Isso foi mostrado pela enorme diferença entre os resultados do pior para o melhor subconjunto (aproximadamente 90%). Isso evidencia a existência de nós críticos para o aprendizado da rede (como mapeado nos experimentos anteriores). Conforme aumentou-se o número de sensores selecionados, aumentou-se também a probabilidade de os nós críticos serem selecionados, o que reduziu a diferença no desempenho entre o melhor e o pior subconjunto.

5.5.4 GraphSAGE-Média

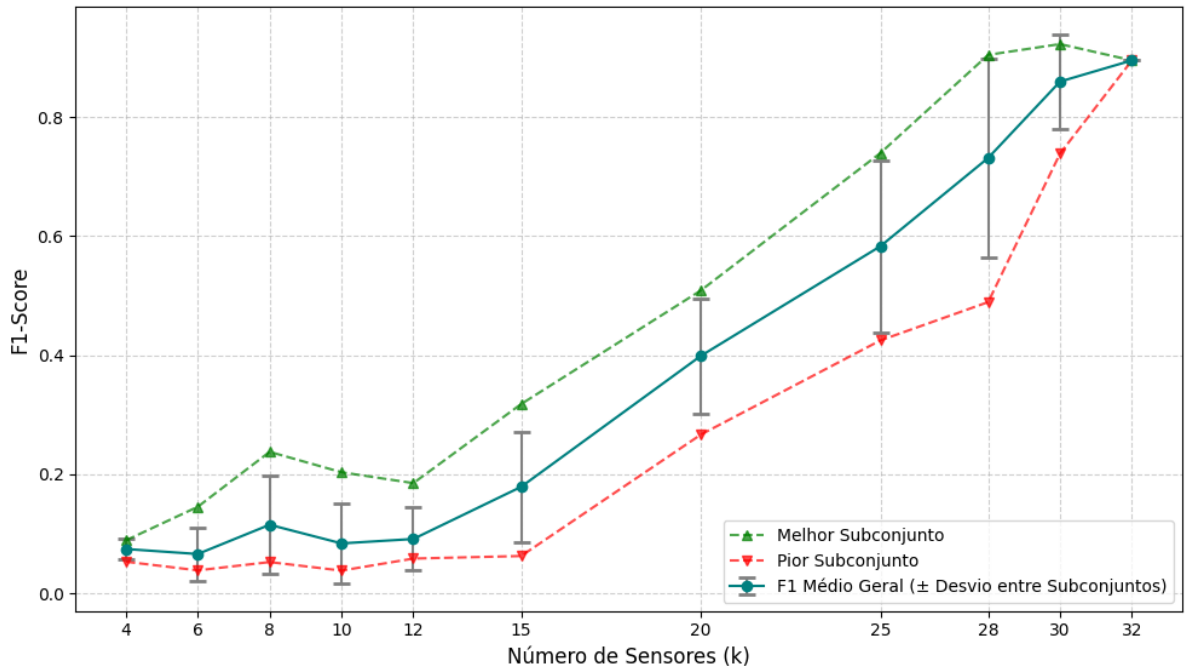
Diferente dos modelos anteriores, após o quantitativo de 15 sensores, este modelo tem uma rampa de aprendizado mais consistente. Entretanto, apesar de uma menor diferença de desempenho alcançado entre os melhores e os piores subconjuntos, essa diferença ainda foi significativa, o que também indica uma dependência topológica dos

Figura 33 – Determinação do número mínimo de sensores para o GraphSAGE-LSTM.



Fonte: Elaborada pelo autor.

Figura 34 – Determinação do número mínimo de sensores para o GraphSAGE-Média.



Fonte: Elaborada pelo autor.

sensores selecionados.

5.6 CONSIDERAÇÕES GERAIS SOBRE OS EXPERIMENTOS ANALISADOS

A execução dos cinco experimentos permitiu uma avaliação detalhada do comportamento das diferentes arquiteturas de GNNs aplicadas à detecção de vazamentos em RDAs. O foco dos experimentos foi identificar as características de cada um dos modelos analisados.

Ao todo, a metodologia exigiu a execução de **2520 treinamentos** completos dos algoritmos a partir do zero. A distribuição ocorreu da seguinte forma: os Experimentos 1, 2 e 3 demandaram 10 treinamentos cada (totalizando 30), guiados pelo vetor de sementes aleatórias (*seeds*) para garantir a reprodutibilidade da condição inicial. O Experimento 4 exigiu mais treinamentos, totalizando 320 execuções, uma vez que o modelo precisou ser retreinado 10 vezes para a avaliação da exclusão individual de cada um dos 32 sensores. Por fim, o Experimento 5 contabilizou 275 treinamentos, decorrentes da análise de 11 quantidades diferentes de sensores, multiplicadas por 5 combinações sorteadas e 5 treinamentos independentes para cada subconjunto.

A arquitetura GGNN, embora não tenha obtido os melhores resultados quando treinada com todos os sensores, mostrou-se mais consistente do que as demais. No experimento 2, apresentou uma queda menos abrupta durante a retirada progressiva de sensores. No experimento 3, foi o único dos modelos a não sofrer com a remoção de sensores individuais. O experimento 4 justificou os resultados dos experimentos anteriores, a GGNN não centraliza seu aprendizado em um único sensor, pelo contrário, distribui sua atenção a todos os sensores da rede, desta forma todos influenciam no processo de aprendizado. Isso ocorre devido ao seu mecanismo de GRU, que consegue inferir o estado de nós inoperantes usando os valores dos sensores vizinhos. Por fim, o experimento 5 mostrou que, apesar dos resultados apresentados no experimento 4, este modelo também possui alguns subconjuntos que proporcionam melhores resultados do que outros. Diante do exposto, conclui-se que este é o modelo mais confiável para operar em condições de falha, mesmo não obtendo o melhor resultado quando todos os sensores estão presentes.

Quanto ao GraphSAGE com o agregador do tipo *polling*, foi o pior dos modelos analisados, sendo ”descartado” após o segundo experimento. Isso se justifica pelo fato de que o operador *Max-Pooling* descarta as quedas de pressão e filtra apenas os picos, perdendo as informações essenciais da rede. As demais arquiteturas de GraphSAGE, tanto a que utiliza como agregador o LSTM quanto a média, apresentaram uma queda acentuada de desempenho perante a remoção progressiva dos sensores (experimento 2). No experimento 3, o modelo com agregador LSTM revelou depender dos dados de todos os sensores pois o agregador decora a estrutura dos dados de uma forma muito rígida e, ao perder qualquer sensor, o modelo entra em colapso global. O modelo com agregador média demonstrou

uma dependência ambígua, em que depende da maioria dos sensores, entretanto alguns sensores fornecem informações totalmente irrelevantes para este modelo. Já no experimento 4, o agregador LSTM mostrou que o seu treinamento depende exclusivamente de poucos sensores da rede e, os demais, praticamente não proporcionam variações quando não estão presentes. O agregador média, por sua vez, revelou-se dependente de apenas alguns sensores e, quando se retiram outros sensores, ao contrário do esperado, ocorre uma melhora no desempenho de treinamento. Essa melhora ocorre devido à eliminação de redundância e ruído topológico. Adicionalmente, no experimento 5, o agregador com LSTM alcançou valores próximos do ideal, com apenas 15 sensores, desde que o subconjunto escolhido contenha os sensores mais críticos. O agregador da média, por sua vez, apresentou uma rampa de crescimento mais consistente, quando comparado com os demais modelos.

A G-LSTM possui uma alta dependência da quantidade de sensores, o que foi evidenciado já no segundo experimento. A G-LSTM também revelou ser altamente dependente de cada um dos sensores da RDA, uma vez que a perda de um único nó quebra a dependência temporal contínua exigida pela célula LSTM. No experimento 4, esta arquitetura teve um comportamento parecido com o GraphSAGE com agregador média, onde a remoção de certos sensores durante o treinamento resultou na melhoria do desempenho. Esta, sem dúvida, foi a descoberta mais contra-intuitiva deste trabalho e pode ser justificada pela eliminação de ruídos topológicos e redundâncias. Por fim, o experimento 5 demonstrou que esta arquitetura também é dependente de alguns sensores críticos e, quando os mesmos estão presentes, é possível obter resultados próximos do ideal com apenas 25 sensores.

Os modelos analisados também mostraram uma não dependência do valor dos sensores próximos aos locais de vazamento. Como as GNNs utilizam o processo de *Message Passing*, após poucas camadas, cada nó compreende a maior parte do grafo. Desta forma, o aprendizado depende dos sensores que oferecem a melhor visão estrutural do sistema , e não do epicentro do vazamento. Intuitivamente, imaginava-se que quanto mais próximo do vazamento, mais fácil seria para o sensor identificar e localizá-lo. Isso pode ser justificado devido ao tempo de aquisição dos dados ser fixo em 10 minutos, ou seja, este intervalo não permite que o modelo capture transientes hidráulicos na transição entre o modelo de operação normal e o início da incidência do vazamento. Assim, os modelos tendem a aprender as diferenças nos valores de pressão provocados quando operam com e sem a incidência de vazamento.

6 CONCLUSÕES

Este estudo investigou diferentes estratégias de *Aprendizado de Máquina* baseadas em *Estruturas de Grafo* para a detecção e localização de vazamentos em Redes de Distribuição de Água (RDAs). O problema dos vazamentos representa uma preocupação global, sendo responsável por perdas econômicas de bilhões de dólares anualmente. No Brasil, a situação é particularmente crítica: estima-se que aproximadamente 40% da água tratada seja perdida ao longo do sistema de distribuição, sendo os vazamentos uma das principais causas desse desperdício.

Com o objetivo de mitigar esse problema, diversos estudos têm sido desenvolvidos visando à criação de técnicas capazes de identificar vazamentos de forma rápida e precisa, com destaque para abordagens baseadas em modelos de *Machine Learning* e *Deep Learning*. Essas técnicas apresentam como principal vantagem a possibilidade de treinamento em ambientes simulados, reduzindo custos operacionais e o tempo necessário para experimentação. Além disso, permitem a geração e avaliação de múltiplos cenários de operação e falha por meio de softwares especializados de simulação hidráulica.

Dentre os programas utilizados para modelagem e simulação hidráulica, destaca-se o EPANET, amplamente adotado, tanto por empresas, como por pesquisadores devido à sua capacidade de realizar simulações hidráulicas realísticas. Com o EPANET, foi possível conduzir simulações variadas, em cenários com ou sem a incidência de vazamentos. A automatização das simulações foi viabilizada pela integração entre o EPANET e o MATLAB, dada a limitação do EPANET para executar rotinas de simulação em períodos prolongados. O processo de validação foi conduzido por experimentos realizados por meio de simulações na RDA da cidade de Hanói, frequentemente utilizada como *benchmark* na literatura.

Foram estruturados cinco experimentos com o intuito de verificar o funcionamento dos modelos em diversas situações. Para isso, utilizou-se a métrica F1-score como parâmetro de desempenho e analisou-se o impacto dos experimentos nos seus valores quando as redes neurais foram treinadas com todos os dados presentes. Os experimentos buscaram analisar os modelos quanto à perda de sensores, sequenciais ou não; importância de cada sensor durante o processo de treinamento e determinação do número mínimo de sensores para que o modelo reporte um bom desempenho.

Com base nos resultados obtidos, conclui-se que a GGNN desponta como a solução mais robusta e viável para aplicações no mundo real. Embora outras arquiteturas possam apresentar picos de precisão marginalmente superiores sob condições ideais, a GGNN demonstrou-se superior quando sujeita à degradação por falha, além de não centralizar

seu aprendizado em alguns poucos sensores. Dentre as redes analisadas, esta característica garante uma melhor redundância operacional, permitindo a detecção de vazamentos mesmo em condições adversas.

Em contrapartida, o GraphSAGE com os agregadores LSTM e média se mostrou altamente sensível a falhas operacionais. Adicionalmente, o modelo que utiliza *polling* como agregador se mostrou incapaz de aprender a localizar vazamentos.

Por fim, o estudo da determinação do número mínimo de sensores mostrou que a localização topológica dos sensores é tão ou mais determinante do que sua quantidade, isso é evidenciado pela distância entre o desempenho dos melhores e piores subconjuntos. Além disso, constatou-se que, para o aprendizado das redes neurais, é mais importante a presença de sensores que ofereçam uma visão estrutural global do sistema do que sensores fisicamente próximos aos epicentros dos vazamentos. Notou-se também que todos os modelos analisados começaram a demonstrar desempenho não aleatório com um subconjunto entre 15 a 20 sensores.

Em síntese, este trabalho evidenciou que a seleção de algoritmo para detecção e localização de vazamentos não deve ser baseada apenas no seu desempenho em condições ideais, mas também na sua resiliência em diversas situações, sejam elas motivadas por falhas ou com intuito de definir o menor número de sensores para que o sistema seja economicamente viável.

Referências

ANGELAKIS LARRY W. MAYS, D. K. N. M. A. N. *Evolution of Water Supply Through the Millennia*. [S.l.]: Iwa Pub, 2012. ISBN 1843395401.

ARAGãO, F. V. *Modelo de maturidade para Smart Cities: uma abordagem multicritério*. Tese (Tese (Doutorado em Engenharia de Produção)) — Universidade Tecnológica Federal do Paraná, Ponta Grossa, November 2020. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/140>.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, IEEE, v. 5, n. 2, p. 157–166, 1994. PMID: 18267787.

CHAN, T. K.; CHIN, C. S.; ZHONG, X. Review of current technologies and proposed intelligent methodologies for water distributed network leakage detection. *IEEE Access*, v. 6, p. 78846–78867, 2018.

COLOMBO, A. F.; LEE, P.; KARNEY, B. W. A selective literature review of transient-based leak detection methods. *Journal of Hydro-environment Research*, Elsevier, v. 2, n. 4, p. 212–227, 2009.

COMMISSION, E.; DIRECTIVE, W. F. *Good Practices on Leakage Management*. [S.l.], 2020. Disponível em: <https://www.eea.europa.eu/publications/good-practices-on-leakage-management>.

CROUCH, D. *Water Management in Ancient Greek Cities*. Oxford University Press, 1993. ISBN 9780199728213. Disponível em: <https://books.google.com.br/books?id=sWBaGj9gmOMC>.

ELIADES, D. G. et al. Epanet-matlab toolkit: An open-source software for interfacing epanet with matlab. In: *Proc. 14th International Conference on Computing and Control for the Water Industry (CCWI)*. The Netherlands: [s.n.], 2016. p. 8.

(EPA), E. P. A. *EPANET 2.0 User's Manual*. 2000.

EREMIA, M.; TOMA, L.; SANDULEAC, M. The smart city concept in the 21st century. *Procedia Engineering*, Elsevier, v. 181, p. 12–19, 2017. ISSN 1877-7058. Disponível em: <https://doi.org/10.1016/j.proeng.2017.02.357>.

FAN, X.; YU, X. B. An innovative machine learning based framework for water distribution network leakage detection and localization. *Structural Health Monitoring*, v. 21, n. 4, p. 1626–1644, 2022. Disponível em: <https://doi.org/10.1177/14759217211040269>.

FAN, X.; ZHANG, X.; YU, B. Machine learning model and strategy for fast and accurate detection of leaks in water supply network. *Journal of Infrastructure Preservation and Resilience*, v. 2, 04 2021.

FAO, IFAD, UNICEF, WFP and WHO. *The State of Food Security and Nutrition in the World 2024*. Roma, Itália: Food and Agriculture Organization of the United

Nations (FAO), 2024. Disponível online. Acesso em: 21 out. 2025. Disponível em: <https://openknowledge.fao.org/items/ebe19244-9611-443c-a2a6-25cec697b361>.

FUJIWARA, O.; KHANG, D. B. A two-phase decomposition method for optimal design of looped water distribution networks. *Water Resources Research*, v. 26, n. 4, p. 539–549, 1990. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/WR026i004p00539>.

GORI, M.; MONFARDINI, G.; SCARSELLI, F. A new model for learning in graph domains. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* [S.l.: s.n.], 2005. v. 2, p. 729–734 vol. 2.

GRANDINI, M.; BAGLI, E.; VISANI, G. *Metrics for Multi-Class Classification: an Overview*. 2020. Disponível em: <https://arxiv.org/abs/2008.05756>.

GéRON, A. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol, CA: O'Reilly Media, 2017. Includes bibliographical references and index. ISBN 978-1491962299.

HAMILTON, W. *Graph Representation Learning*. Morgan & Claypool Publishers, 2020. (Synthesis lectures on artificial intelligence and machine learning). ISBN 9781681739632. Disponível em: <https://books.google.com.br/books?id=V6HnzQEACAAJ>.

HAMILTON, W. L.; YING, R.; LESKOVEC, J. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017. Disponível em: <http://arxiv.org/abs/1706.02216>.

HAMILTON, W. L.; YING, R.; LESKOVEC, J. Representation learning on graphs: Methods and applications. *CoRR*, abs/1709.05584, 2017. Disponível em: <http://arxiv.org/abs/1709.05584>.

HE, K. et al. Identity mappings in deep residual networks. *uropean Conference*, 2016. Disponível em: <https://arxiv.org/abs/1603.05027>.

HIRNER, W.; LAMBERT, A. Losses from water supply systems : standard terminology and recommended performance measures. *International Water Association*, p. 13, 10 2000.

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

HOSSIN, M.; SULAIMAN, M. N. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 2, p. 1, 2015.

JOHNSON, D. D. Learning graphical state transitions. In: *International Conference on Learning Representations*. [S.l.: s.n.], 2016.

KIPF, T. N.; WELLING, M. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017.

LI, Y. et al. *Gated Graph Sequence Neural Networks*. arXiv, 2015. Disponível em: <https://arxiv.org/abs/1511.05493>.

LIU, J. Z. Z. *Introduction to Graph Neural Networks*. Morgan Claypool, 2020. (Synthesis Lectures on Artificial Intelligence and Machine Learning). ISBN 1681737655, 9781681737652. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=F2A78B373081C648CEB4350817145DE1>.

LIU, Y. et al. Water pipeline leakage detection based on machine learning and wireless sensor networks. *Sensors*, v. 19, n. 23, 2019. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/19/23/5086>.

LOUREIRO, D. et al. Water distribution systems flow monitoring and anomalous event detection: A practical approach. *Urban Water Journal*, v. 13, p. 1–11, 01 2015.

LUO, J. et al. A transformer-based approach to leakage detection in water distribution networks. *Sensors*, MDPI, v. 24, n. 19, p. 6294, 2024.

LUčIN, I. et al. Data-driven leak localization in urban water distribution networks using big data for random forest classifier. *Mathematics*, v. 9, n. 6, 2021. ISSN 2227-7390. Disponível em: <https://www.mdpi.com/2227-7390/9/6/672>.

MASHHADI, N. et al. Use of machine learning for leak detection and localization in water distribution systems. *Smart Cities*, v. 4, n. 4, p. 1293–1315, 2021. ISSN 2624-6511. Disponível em: <https://www.mdpi.com/2624-6511/4/4/69>.

MATTOS DANIEL VALENCIA CáRDENAS, K. d. B. M. M. A. Equipe Técnica: Caio Petrillo Vieira de. *Diagnóstico Temático: Serviços de Água e Esgoto – Visão Geral*. Brasília, DF, Brasil, 2023. Acesso em: 21 out. 2025. Disponível em: <https://www.gov.br/cidades/pt-br/acesso-a-informacao/acoes-e-programas/saneamento/snis/produtos-do-snis/diagnosticos/DIAGNOSTICO_TEMATICO_VISAO_GERAL_AE_SNIS_2023.pdf>.

MAYS, L.; KOUTSOYIANNIS, D.; ANGELAKIS, A. A brief history of urban water supply in antiquity. *Water Supply*, v. 7, n. 1, p. 1–12, 03 2007. ISSN 1606-9749. Disponível em: <https://doi.org/10.2166/ws.2007.001>.

MIENYE, I. D.; SWART, T. G.; OBAIDO, G. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, v. 15, n. 9, 2024. ISSN 2078-2489. Disponível em: <https://www.mdpi.com/2078-2489/15/9/517>.

National Geographic Society. *Roman Aqueducts*. Washington, DC, EUA: National Geographic Society, 2025. Disponível online. Acessed: 2024-11-01. Disponível em: <https://education.nationalgeographic.org/resource/roman-aqueducts/>.

NKEMENI, V.; MIEYEVILLE, F.; TSAFACK, P. A distributed computing solution based on distributed kalman filter for leak detection in wsn-based water pipeline monitoring. *Sensors*, v. 20, n. 18, 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/18/5204>.

PONCE, M. V. C.; CASTAñóN, L. E. G.; CAYUELA, V. P. Model-based leak detection and location in water distribution networks considering an extended-horizon analysis of pressure sensitivities. *Journal of Hydroinformatics*, v. 16, n. 3, p. 649–670, 11 2013. ISSN 1464-7141. Disponível em: <https://doi.org/10.2166/hydro.2013.019>.

PUUST, R. et al. A review of methods for leakage management in pipe networks. *Urban Water Journal*, Taylor Francis, v. 7, n. 1, p. 25–45, 2010. Disponível em: <https://doi.org/10.1080/15730621003610878>.

PéREZ, R. et al. Methodology for leakage isolation using pressure sensitivity analysis in water distribution networks. *Control Engineering Practice*, v. 19, n. 10, p. 1157–1167, 2011. ISSN 0967-0661. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0967066111001201>.

QI, S. et al. Research on an optimized leakage locating model in water distribution system. *Procedia Engineering*, v. 89, p. 1569–1576, 2014. ISSN 1877-7058. 16th Water Distribution System Analysis Conference, WDSA2014. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877705814025727>.

QUINONES-GRUEIRO, M. et al. Comparison of classifiers for leak location in water distribution networks. *IFAC-PapersOnLine*, v. 51, p. 407–413, 01 2018 (1).

QUIñONES-GRUEIRO, M. et al. An unsupervised approach to leak detection and location in water distribution networks. *International Journal of Applied Mathematics and Computer Science*, v. 28, n. 2, p. 283–295, 2018 (2). Disponível em: <https://doi.org/10.2478/amcs-2018-0020>.

RASHID, S.; AKRAM, U.; KHAN, S. A. Wml: Wireless sensor network based machine learning for leakage detection and size estimation. *Procedia Computer Science*, v. 63, p. 171–176, 2015. ISSN 1877-0509. The 6th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2015)/ The 5th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2015)/ Affiliated Workshops. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050915024643>.

RODRIGUES, W. C.; ROLLE, R. P.; GODOY, E. P. Localização de vazamentos em redes de distribuição de água baseada em amostragem e agregação de grafos (graphsage). In: *Anais do XXV Congresso Brasileiro de Automática (CBA 2024)*. São José do Rio Preto, SP: SBA, 2024. p. 1437–1443.

RODRÍGUEZ-ARGOTE, C. A. et al. Two-leak isolation in water distribution networks based on k-nn and linear discriminant classifiers. *Water*, MDPI, v. 15, n. 17, p. 3090, 2023.

ROGERS, D. Leaking water networks: An economic and environmental disaster. *Procedia Engineering*, v. 70, p. 1421–1429, 2014. ISSN 1877-7058. 12th International Conference on Computing and Control for the Water Industry, CCWI2013. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877705814001593>.

ROJEK, I.; STUDZINSKI, J. Detection and localization of water leaks in water nets supported by an ict system with artificial intelligence methods as a way forward for smart cities. *Sustainability*, v. 11, n. 2, 2019. ISSN 2071-1050. Disponível em: <https://www.mdpi.com/2071-1050/11/2/518>.

ROLLE, R. P. *Detecção e localização de vazamentos utilizando Internet das Coisas e Aprendizado de Máquina*. Tese (Tese (Doutorado em Engenharia Elétrica)) — Faculdade de Engenharia, Universidade Estadual Paulista "Júlio de Mesquita Filho", Bauru, 2024. Orientador: Eduardo Paciência Godoy. Disponível em: <https://hdl.handle.net/11449/254317>.

ROLLE, R. P. et al. Data-driven leak detection and localization using LPWAN and Deep Learning. In: IEEE. *2022 IEEE International Workshop on Metrology for*

Industry 4.0 & IoT (MetroInd4.0&IoT). 2022. p. 403–407. Disponível em: <http://dx.doi.org/10.1109/MetroInd4.0IoT54413.2022.9831619>.

ROLLE, R. P. et al. Detecção de vazamentos em redes hidráulicas utilizando redes neurais de grafos. In: *Anais do XXIV Congresso Brasileiro de Automática (CBA 2022)*. Fortaleza, CE: SBA, 2022. v. 3, n. 1, p. 3769–3776.

ROLLE, R. P. et al. Leveraging graph-based leak localization in water distribution networks. In: *2024 IEEE International Workshop on Metrology for Industry 4.0 IoT (MetroInd4.0 IoT)*. [S.l.: s.n.], 2024. p. 192–197.

ROSSMAN, L. et al. *EPANET 2.2 User Manual*. [S.l.], 2020. Disponível em: <https://19january2021snapshot.epa.gov/sites/static/files/2020-05/documents/epanet_userss_manual_2.2.0.pdf>.

SANCHEZ-LENGELING, B. et al. *A Gentle Introduction to Graph Neural Networks*. 2021. Https://distill.pub/2021/gnn-intro.

SCARSELLI, F. et al. The graph neural network model. *IEEE Transactions on Neural Networks*, v. 20, n. 1, p. 61–80, 2009.

SCOTT, S. L.; BARRUFET, M. A. *Worldwide Assessment of Industry Leak Detection Capabilities for Single & Multiphase Pipelines*. College Station, TX, USA, 2003. Disponível em: <https://otrc.tamu.edu/research/publications/subsea-systems/worldwide-assessment-of-industry-leak-detection-capabilities-for-single-multiphase-pipelines/a120/>.

SHRAVANI, D. et al. A machine learning approach to water leak localization. In: *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*. [S.l.: s.n.], 2019. p. 1–6.

SOLDEVILA, A. et al. Leak localization in water distribution networks using a mixed model-based/data-driven approach. *Control Engineering Practice*, v. 55, p. 162–173, 2016. ISSN 0967-0661. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0967066116301526>.

SUN, C. et al. Leak localization in water distribution networks using pressure and data-driven classifier approach. *Water*, v. 12, n. 1, 2020. ISSN 2073-4441. Disponível em: <https://www.mdpi.com/2073-4441/12/1/54>.

TALARI, S. et al. A review of smart cities based on the internet of things concept. *Energies*, MDPI, v. 10, n. 4, p. 421, 2017. ISSN 1996-1073. Disponível em: <https://www.mdpi.com/1996-1073/10/4/421>.

TEJASWINI, M.; SILO, G.; ASHWINI, B. Design of water supply distribution network using epanet software: a case study of kuvempu nagar, mandya. *International Research Journal of Engineering and Technology*, v. 7, p. 1582, 2020.

TOMAZINI, L. et al. Graphleak: A realistic dataset for analyzing leaks in water distribution systems. *Congresso Brasileiro de Automática*, 2024. Accessed: 2024-10-24. Disponível em: <https://www.sba.org.br/cba2024/papers/paper_7042.pdf>.

TORRES, L. et al. Kalman filters for leak diagnosis in pipelines: Brief history and future research. *Journal of Marine Science and Engineering*, v. 8, p. 173, 03 2020.

WALSKI, T. M. A history of water distribution. *Journal AWWA*, v. 98, n. 3, p. 110–121, 2006. Disponível em: <https://awwa.onlinelibrary.wiley.com/doi/abs/10.1002/j.1551-8833.2006.tb07611.x>.

WALSKI, T. M. et al. *Advanced Water Distribution Modeling and Management*. [S.l.]: Haestad Press, 2003.

XU, K. et al. How powerful are graph neural networks? *ArXiv*, abs/1810.00826, 2019.