



UNIVERSIDADE ESTADUAL PAULISTA

“JÚLIO DE MESQUITA FILHO”

Faculdade de Engenharia de Bauru

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

WELITON DO CARMO RODRIGUES

**Localização de Vazamentos em Redes de Distribuição de Água  
baseada em Amostragem e Agregação de Grafos (GraphSAGE)**

Bauru

2025

WELITON DO CARMO RODRIGUES

**Localização de Vazamentos em Redes de Distribuição de Água  
baseada em Amostragem e Agregação de Grafos (GraphSAGE)**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Estadual Paulista “Júlio de Mesquita Filho” para a Defesa de Mestrado em Engenharia Elétrica.

**Área de concentração:** Automação  
**Linha de pesquisa:** Mecatrônica

**Orientador (a):** Prof. Dr. Eduardo Paciência Godoy  
**Coorientador (a):** Prof. Dr. Rodrigo Pita Rolle

Bauru  
2025

R696l      Rodrigues, Weliton do Carmo  
Localização de Vazamentos em Redes de Distribuição de Água  
baseada em Amostragem e Agregação de Grafos (GraphSAGE) /  
Weliton do Carmo Rodrigues. -- Bauru, 2025  
63 p.  
  
Dissertação (mestrado) - Universidade Estadual Paulista (UNESP),  
Faculdade de Engenharia, Bauru  
Orientador: Eduardo Paciência Godoy  
Coorientador: Rodrigo Pita Rolle  
  
1. Localização de Vazamentos. 2. Redes de Distribuição de Água.  
3. Redes Neurais de Grafos. 4. Inteligência Artificial. I. Título.

**ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE WELITON DO CARMO RODRIGUES, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, DA FACULDADE DE ENGENHARIA - CÂMPUS DE BAURU.**

Aos 07 dias do mês de agosto do ano de 2025, às 14h, por meio de Videoconferência, realizou-se a defesa de DISSERTAÇÃO DE MESTRADO de WELITON DO CARMO RODRIGUES, intitulada **LOCALIZAÇÃO DE VAZAMENTOS EM REDES DE DISTRIBUIÇÃO DE ÁGUA BASEADA EM AMOSTRAGEM E AGREGAÇÃO DE GRAFOS (GRAPHSAGE)**. A Comissão Examinadora foi constituída pelos seguintes membros: Prof. Dr. EDUARDO PACIÊNCIA GODOY (Orientador(a) - Participação Virtual) do(a) Departamento de Engenharia de Controle e Automação / Instituto de Ciência e Tecnologia UNESP Câmpus de Sorocaba, Prof. Dr. IVANOVITCH MEDEIROS DANTAS DA SILVA (Participação Virtual) do(a) Departamento de Engenharia de Computação e Automação / Universidade Federal do Rio Grande do Norte, Prof. Dr. LEOPOLDO ANDRÉ DUTRA LUSQUINO FILHO (Participação Virtual) do(a) Engenharia de Controle e Automação / ICTS - Instituto de Ciência e Tecnologia de Sorocaba - UNESP. Após a exposição pelo mestrando e arguição pelos membros da Comissão Examinadora que participaram do ato, de forma presencial e/ou virtual, o discente recebeu o conceito final: *Aprovado*. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelo(a) Presidente(a) da Comissão Examinadora.

Prof. Dr. EDUARDO PACIÊNCIA GODOY



UNIVERSIDADE ESTADUAL PAULISTA  
"JÚLIO DE MESQUITA FILHO"  
ICPEdu - Certificado Pessoal  
<https://pessoal.icpedu.rnp.br/public/verificar-assinatura>

Assinado de forma  
digital por Eduardo  
Paciencia  
Godoy:29137583808  
Dados: 2025.08.07  
16:37:30 -03'00'

## **IMPACTO POTENCIAL DESSA PESQUISA**

A presente pesquisa representa um avanço significativo no campo da gestão de redes de distribuição de água ao propor e validar uma abordagem inovadora baseada em redes neurais de grafos (GraphSAGE) para a detecção e localização de vazamentos. A metodologia se destaca por incorporar uma matriz de ponderação baseada na distância física entre os nós da rede, o que permite uma representação mais realista das características hidráulicas do sistema, contribuindo para o aumento da acurácia do modelo. Esta inovação tecnológica possui um alto potencial de impacto social, econômico e ambiental, visto que vazamentos em Redes de Distribuição de Água (RDAs) representam uma das principais causas de perdas no abastecimento urbano, comprometendo a eficiência operacional e gerando impactos financeiros e ambientais significativos. Com aplicações práticas viáveis, inclusive em cenários com recursos limitados, esta solução pode auxiliar companhias de saneamento a tomarem decisões mais precisas e eficientes. Os resultados empíricos da pesquisa demonstram que o uso do algoritmo GraphSAGE, especialmente com o agregador LSTM e a matriz ponderada por distâncias, apresenta melhorias significativas na acurácia da detecção e localização de vazamentos, superando o algoritmo GGNN utilizado como referência. A contribuição desta pesquisa para a redução do desperdício de água alinha-se diretamente com o Objetivo de Desenvolvimento Sustentável (ODS) 6 das Nações Unidas, que visa assegurar a disponibilidade e a gestão sustentável da água e saneamento para todos. Ao otimizar a identificação de perdas, o trabalho fortalece a resiliência e a sustentabilidade de sistemas urbanos, promovendo uma gestão hídrica mais eficiente e inteligente.

## **POTENTIAL IMPACT OF THIS RESEARCH**

This research represents a significant advancement in the field of water distribution network management by proposing and validating an innovative approach based on graph neural networks (GraphSAGE) for leak detection and localization. The methodology stands out by incorporating a weighting matrix based on the physical distance between network nodes, which allows for a more realistic representation of the system's hydraulic characteristics, contributing to increased model accuracy. This technological innovation has high potential for social, economic, and environmental impact, as leaks in Water Distribution Networks (WDNs) represent one of the main causes of losses in urban supply, compromising operational efficiency and generating significant financial and environmental impacts. With viable practical applications, even in resource-limited scenarios, this solution can assist water utilities in making more precise and efficient decisions. The empirical results of the research demonstrate that the use of the GraphSAGE algorithm, especially with the LSTM aggregator and the distance-weighted matrix, shows significant improvements in the accuracy of leak detection and localization, surpassing the GGNN algorithm used as a benchmark. This research's contribution to reducing water waste directly aligns with the United Nations' Sustainable Development Goal (SDG) 6, which aims to ensure the availability and sustainable management of water and sanitation for all. By optimizing leak identification, the work strengthens the resilience and sustainability of urban systems, promoting more efficient and intelligent water management.

## AGRADECIMENTOS

Agradeço primeiramente a Deus, por me sustentar em cada etapa desta jornada, iluminando meus caminhos com sabedoria, força e perseverança. À minha esposa, **Ariadne Rodrigues**, pelo amor, companheirismo e incentivo incondicional. Sua paciência, apoio constante e compreensão foram fundamentais para que eu pudesse me dedicar plenamente a esta etapa da minha vida acadêmica. Ao meu orientador, **Prof. Dr. Eduardo Paciência Godoy**, por sua orientação competente, pela confiança no meu trabalho e pelo comprometimento com a minha formação. Sua dedicação e visão crítica foram essenciais para o desenvolvimento desta pesquisa. Ao meu coorientador, **Prof. Dr. Rodrigo Pita Rolle**, pela valiosa colaboração, pelas contribuições técnicas e pela disponibilidade ao longo de todo o projeto. Seu apoio e conhecimento ampliaram significativamente a qualidade deste trabalho. Aos **professores membros da banca examinadora**, pelos comentários, sugestões e contribuições que enriqueceram este trabalho e contribuíram para o meu crescimento acadêmico e pessoal. Agradeço pela atenção e generosidade em dedicar seu tempo à avaliação desta pesquisa. Agradeço à **CAPES**, pelo apoio financeiro concedido por meio da bolsa de estudos, que viabilizou a realização desta pesquisa. Este projeto foi financiado, em parte, pelo **Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)**, sob os contratos **142383/2019-8** e **303967/2021-8**, aos quais também expresso meus sinceros agradecimentos. A todos que, de alguma forma, contribuíram para esta conquista, deixo aqui o meu muito obrigado.

## CERTIFICADO DE APROVAÇÃO

**Título da Dissertação de Mestrado:** LOCALIZAÇÃO DE VAZAMENTOS EM REDES DE DISTRIBUIÇÃO DE ÁGUA BASEADA EM AMOSTRAGEM E AGREGAÇÃO DE GRAFOS (GRAPHSAGE)

**Aluno:** Weliton do Carmo Rodrigues

**Orientador:** Eduardo Paciencia Godoy

**Coorientador:** Rodrigo Pita Rolle

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica, Programa De Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia - Câmpus De Bauru, pela Comissão Examinadora:

Prof. Dr. EDUARDO PACIÊNCIA GODOY (Orientador)  
Departamento de Engenharia de Controle e  
Automação / Universidade Estadual Paulista  
Câmpus de Sorocaba

 Assinado de forma digital por Eduardo Paciencia Godoy:29137583808  
UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"  
ICPEdu - Certificado Pessoal  
Dados: 2025.08.07 14:32:38 -03'00'  
<https://pessoal.icpedu.rnp.br/public/verificar-assinatura>

Prof. Dr. IVANOVITCH MEDEIROS DANTAS DA SILVA  
Departamento de Engenharia de Computação e  
Automação / Universidade Federal do Rio Grande do Norte

Documento assinado digitalmente  
 IVANOVITCH MEDEIROS DANTAS DA SILVA  
Data: 07/08/2025 17:21:23-0300  
Verifique em <https://validar.iti.gov.br>

Prof. Dr. LEOPOLDO ANDRÉ DUTRA LUSQUINO FILHO  
Departamento de Engenharia de Controle e Automação  
/ Universidade Estadual Paulista Câmpus de Sorocaba

Documento assinado digitalmente  
 LEOPOLDO ANDRÉ DUTRA LUSQUINO FILHO  
Data: 08/08/2025 11:05:15-0300  
Verifique em <https://validar.iti.gov.br>

Bauru, 07 de agosto de 2025.

## RESUMO

A rede de distribuição de água (RDA) desempenha um papel essencial na garantia do acesso confiável à água potável em comunidades urbanas e rurais. Contudo, o desperdício de água resultante de vazamentos na infraestrutura representa um desafio significativo, comprometendo a eficiência operacional e gerando impactos ambientais e financeiros. A detecção eficaz de vazamentos é uma tarefa complexa que demanda uma análise adequada dos dados, especialmente considerando as correlações existentes entre eles. Nesse contexto, a modelagem baseada em grafos proporciona uma representação robusta da estrutura das RDAs, enquanto as redes neurais de grafos (*Graph Neural Networks* – GNNs) apresentam o potencial de melhorar a precisão na detecção e localização de vazamentos. Este trabalho busca dar continuidade aos estudos sobre a aplicação de GNNs na localização de vazamentos em RDAs, propondo a utilização do algoritmo de Amostragem e Agregação de Grafos (*Graph Sample and Aggregation* – GraphSAGE) para essa tarefa. Além disso, investiga-se o uso de uma matriz de ponderação baseada nas distâncias físicas entre os nós da rede, em substituição à tradicional matriz de adjacência binária, com o objetivo de representar de forma mais realista as relações estruturais da RDA e aprimorar o aprendizado do modelo. Para a avaliação da proposta, foi utilizado como referência um estudo anterior que demonstrou a viabilidade da modelagem em grafos para essa finalidade, servindo como base comparativa para os resultados obtidos com o GraphSAGE.

**Palavras-chave:** Detecção de Vazamentos, Modelagem Baseada em Grafos, Redes Neurais de Grafo (GNN), Simulação de Redes de Distribuição de Água.

## ABSTRACT

The water distribution network (WDN) plays an essential role in ensuring reliable access to potable water in both urban and rural communities. However, water loss caused by infrastructure leaks poses a significant challenge, compromising operational efficiency and generating environmental and financial impacts. Effective leak detection is a complex task that requires proper data analysis, especially considering the correlations among data points. In this context, graph-based modeling offers a robust representation of the structure of WDNs, while Graph Neural Networks (GNNs) have the potential to improve the accuracy of leak detection and localization. This work aims to advance studies on the application of GNNs for leak localization in WDNs by proposing the use of the Graph Sample and Aggregation (GraphSAGE) algorithm for this task. In addition, it investigates the use of a weighting matrix based on the physical distances between network nodes, as an alternative to the traditional binary adjacency matrix, with the goal of more realistically representing the structural relationships within the WDN and enhancing the model's learning process. To evaluate the proposed approach, a previous study that demonstrated the feasibility of graph-based modeling for this purpose was used as a reference, serving as a benchmark for the results obtained with GraphSAGE.

**Keywords:** Leakage Detection, Graph-Based Modeling, Graph Neural Networks (GNN), Water Distribution Network Simulation.

## LISTA DE ABREVIATURAS E SIGLAS

|           |                                                      |
|-----------|------------------------------------------------------|
| AE        | <i>Autoencoders</i>                                  |
| ANN       | <i>Artificial Neural Network</i>                     |
| CFVT      | Circuito Fechado de Televisão                        |
| CNN       | <i>Convolutional Neural Network</i>                  |
| EPA       | <i>United States Environmental Protection Agency</i> |
| GAT       | <i>Graph Attention Network</i>                       |
| GCN       | <i>Graph Convolutional Network</i>                   |
| GraphSAGE | <i>Graph Sample and Aggregation</i>                  |
| GNN       | <i>Graph Neural Network</i>                          |
| GGNN      | <i>Gated Graph Neural Network</i>                    |
| GRNN      | <i>Graph Recurrent Neural Network</i>                |
| IoT       | <i>Internet of Things</i>                            |
| LDA       | <i>Linear Discriminant Analise</i>                   |
| LPM       | Litros por Minuto                                    |
| LSTM      | <i>Long Short-Term Memory</i>                        |
| mca       | Metros de coluna de água                             |
| ML        | <i>Machine Learning</i>                              |
| ODS 6     | Objetivo de Desenvolvimento Sustentável 6            |
| ONU       | Organização das Nações Unidas                        |
| RDA       | Rede de Distribuição de Água                         |
| ReLU      | <i>Rectified Linear Unit</i>                         |
| RNA       | Redes Neurais Artificiais                            |
| RNF       | Reservatório de Nível Fixo                           |
| RNV       | Reservatório de Nível Variável                       |
| VS Code   | <i>Visual Studio Code</i>                            |
| WL        | Weisfeiler-Lehman                                    |

## LISTA DE FIGURAS

|                                                                                                                                                                                                                 |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 - Panorama das técnicas empregadas na detecção de vazamentos em RDAs.....                                                                                                                              | 5  |
| Figura 2 – Exemplo de redes de distribuição .....                                                                                                                                                               | 14 |
| Figura 3 – Padrão de consumo em uma RDA para o intervalo 24 horas .....                                                                                                                                         | 17 |
| Figura 4 - Ilustração visual da abordagem de amostragem e agregação do GraphSAGE .....                                                                                                                          | 21 |
| Figura 5 - Ilustração de uma RDA hipotética representando um condomínio fictício com quatro casas.....                                                                                                          | 31 |
| Figura 6 - Fluxograma da rotina de simulação implementada utilizando o EPANET-MATLAB <i>toolkit</i> .....                                                                                                       | 32 |
| Figura 7 - Conversão da estrutura da RDA em grafo .....                                                                                                                                                         | 34 |
| Figura 8 - Exemplo de grafo com suas matrizes de adjacência e distâncias .....                                                                                                                                  | 35 |
| Figura 9 - Fluxograma de aplicação do modelo de GNN aplicado para detecção e localização de vazamentos.....                                                                                                     | 35 |
| Figura 10 - Fluxograma final da aplicação do modelo de GNN aplicado para detecção e localização de vazamentos.....                                                                                              | 39 |
| Figura 11 - RDA do primeiro cenário. ....                                                                                                                                                                       | 42 |
| Figura 12 – Grafo equivalente à RDA do primeiro cenário. ....                                                                                                                                                   | 43 |
| Figura 13 - Distribuição da acurácia de validação dos modelos GGNN e GraphSAGE (Mean, Pooling e LSTM) com matriz de adjacência binária e matriz ponderada por distâncias, para o Cenário 1.....                 | 46 |
| Figura 14 - RDA do segundo cenário.....                                                                                                                                                                         | 47 |
| Figura 15 - Grafo equivalente à RDA do segundo cenário. ....                                                                                                                                                    | 48 |
| Figura 16 - Distribuição da acurácia de validação dos modelos GGNN e GraphSAGE ( <i>Mean</i> , <i>Pooling</i> e LSTM) com matriz de adjacência binária e matriz ponderada por distâncias, para o Cenário 2..... | 50 |

## LISTA DE TABELAS

|                                                                                                                 |    |
|-----------------------------------------------------------------------------------------------------------------|----|
| Tabela 1 – Parâmetro comuns dos modelos de RDAs gerados .....                                                   | 32 |
| Tabela 2 – Atribuições de rótulos para as classes de vazamentos do cenário 1.....                               | 44 |
| Tabela 3 – Matriz de adjacência binária do Cenário 1. ....                                                      | 44 |
| Tabela 4 – Matriz de distâncias do Cenário 1. ....                                                              | 45 |
| Tabela 5 - Atribuições de rótulos para as classes de vazamentos do cenário 2.....                               | 49 |
| Tabela 6 – Apresentação das métricas de avaliação dos modelos de GNNs aplicados na tarefa para o Cenário 1..... | 51 |
| Tabela 7 - Apresentação das métricas de avaliação dos modelos de GNNs aplicados na tarefa para o Cenário 2..... | 52 |
| Tabela 8 - Matriz de adjacência binária do Cenário 2.....                                                       | 62 |
| Tabela 9 - Matriz de ponderação baseada nas distâncias do Cenário 2.....                                        | 63 |

## Sumário

|                                                                            |           |
|----------------------------------------------------------------------------|-----------|
| <b>1. INTRODUÇÃO</b>                                                       | <b>1</b>  |
| 1.1. Apresentação do tema da pesquisa                                      | 1         |
| 1.2. Objetivo da pesquisa                                                  | 2         |
| 1.3. Estrutura e conteúdo abordado                                         | 3         |
| <b>2. REVISÃO BIBLIOGRÁFICA</b>                                            | <b>5</b>  |
| 2.1. Detecção e localização de vazamentos                                  | 5         |
| 2.1.1. Métodos de detecção passiva                                         | 5         |
| 2.1.2. Métodos de detecção ativa                                           | 7         |
| 2.1.3. Aprendizado de máquina para detecção de vazamentos                  | 8         |
| 2.1.4. Aplicação baseada em grafos                                         | 10        |
| 2.1.5. Uso do GraphSAGE para Detecção de Anomalias                         | 11        |
| <b>3. FUNDAMENTAÇÃO TEÓRICA</b>                                            | <b>13</b> |
| 3.1. Modelagem de redes de distribuição de água                            | 13        |
| 3.1.1. Componentes de um modelo                                            | 15        |
| 3.1.2. Lei da conservação de massa                                         | 15        |
| 3.1.3. Lei da conservação de energia                                       | 16        |
| 3.1.4. Consumo de água                                                     | 16        |
| 3.1.5. Perdas de carga                                                     | 17        |
| 3.1.6. Controles                                                           | 18        |
| 3.1.7. Modelagem de vazamentos                                             | 18        |
| 3.2. Redes neurais de grafo                                                | 19        |
| 3.2.1. Arquitetura e funcionamento do GraphSAGE                            | 20        |
| 3.2.1.1. Algoritmos de geração de <i>embeddings</i>                        | 21        |
| 3.2.1.2. Aprendizado dos parâmetros do GraphSAGE                           | 25        |
| 3.2.1.3. Arquitetura de agregadores                                        | 25        |
| 3.3. Representação de grafos: matrizes de adjacência e matrizes ponderadas | 27        |
| 3.3.1. Matriz de adjacência (binária)                                      | 27        |
| 3.3.2. Matrizes ponderadas                                                 | 28        |
| <b>4. MATERIAIS E MÉTODOS</b>                                              | <b>29</b> |
| 4.1. Simulação de cenários via EPANET                                      | 29        |
| 4.1.1. Configuração da rede de distribuição de água                        | 29        |
| 4.1.2. Aquisição dos dados                                                 | 31        |
| 4.1.3. Geração dos cenários                                                | 32        |
| 4.1.4. Conversão da estrutura da RDA em grafo                              | 33        |

|        |                                                     |    |
|--------|-----------------------------------------------------|----|
| 4.1.5. | Fluxograma da aplicação do modelo de GNN .....      | 35 |
| 4.1.6. | Processamento de dados.....                         | 37 |
| 4.1.7. | Parâmetros utilizados para os algoritmos .....      | 38 |
| 4.2.   | Métodos de avaliação da performance do modelo ..... | 38 |
| 5.     | RESULTADOS E DISCUSSÃO .....                        | 41 |
| 5.1.   | Descrição dos experimentos .....                    | 41 |
| 5.2.   | Cenário 1 .....                                     | 41 |
| 5.2.1. | Performance dos modelos para o Cenário 1 .....      | 45 |
| 5.3.   | Cenário 2 .....                                     | 47 |
| 5.3.1. | Performance dos modelos para o Cenário 2 .....      | 50 |
| 5.4.   | Considerações sobre os resultados .....             | 51 |
| 6.     | CONCLUSÃO.....                                      | 55 |
|        | REFERÊNCIAS.....                                    | 57 |
|        | APÊNDICES.....                                      | 62 |

## 1. INTRODUÇÃO

### 1.1. Apresentação do tema da pesquisa

A rede de distribuição de água (RDA) é essencial para a sustentabilidade de comunidades urbanas e rurais, fornecendo acesso seguro e confiável à água potável. A gestão sustentável dessas redes deve priorizar a acessibilidade, a qualidade, a quantidade e a confiabilidade da água (Velayudhan et al., 2022). No entanto, o desperdício causado por vazamentos na infraestrutura representa um desafio significativo, comprometendo a eficiência operacional e gerando impactos ambientais e financeiros consideráveis.

Alinhado ao Objetivo de Desenvolvimento Sustentável 6 (ODS 6) das Nações Unidas, que visa assegurar a disponibilidade e a gestão sustentável da água e do saneamento para todos, o enfrentamento das perdas em redes de distribuição tornou-se uma prioridade global. A ONU destaca que a adoção de tecnologias para detecção de vazamentos é essencial para aumentar a eficiência no uso da água, promover a sustentabilidade dos sistemas de abastecimento e fortalecer a governança dos recursos hídricos (UN-Water, 2024). Tais medidas contribuem diretamente para reduzir desperdícios, ampliar o acesso à água potável e apoiar a resiliência das infraestruturas hidráulicas, especialmente em áreas urbanas em expansão.

Entretanto, o desenvolvimento de sistemas eficazes para a detecção e localização de vazamentos em RDAs é uma tarefa desafiadora. A análise adequada dos dados provenientes de experimentos físicos ou simulações é fundamental, pois desconsiderar as correlações entre os dados pode comprometer a confiabilidade dos sistemas de detecção e afetar negativamente a precisão na localização dos vazamentos. Nesse contexto, a modelagem baseada em grafos oferece uma forma robusta de representar a estrutura e a dinâmica das RDAs, contribuindo para aumentar a eficiência na detecção de vazamentos (Mpiana et al., 2017). Mapear as interconexões entre componentes do sistema, como tubulações, válvulas e pontos de monitoramento, permite uma compreensão mais detalhada e precisa da dinâmica do sistema, facilitando uma representação visual e estrutural de redes complexas que preserva suas características e relações.

A aplicação de técnicas de aprendizado de máquina para a detecção de vazamentos em RDAs tem o potencial de melhorar a precisão e a eficiência dos sistemas já existentes (Kang et al., 2018). Integrar a capacidade de aprendizado das redes neurais à estrutura visual e relacional dos grafos utilizados na modelagem das RDAs potencializa a identificação precisa de vazamentos.

As redes neurais de grafos (*Graph Neural Networks* – GNNs) (Scarselli et al., 2009) são modelos de aprendizado de máquina que utilizam estruturas de grafos para representar e processar informações (Li et al., 2023). Compostas por nós que representam entidades ou conceitos e por arestas que indicam as relações entre esses elementos, as GNNs oferecem uma representação rica em dados interconectados, sendo particularmente adequadas para lidar com dados não euclidianos, como os presentes nas RDAs.

A representação em grafo das RDAs permite uma compreensão detalhada das interconexões entre os componentes do sistema, enquanto as GNNs possibilitam a análise e o aprendizado de padrões complexos de dados correlacionados, como variações de pressão, fluxo e vazão. A combinação dessas abordagens proporciona uma estrutura robusta para identificar e localizar vazamentos, promovendo maior precisão e eficiência, ao capitalizar as vantagens tanto dos grafos na modelagem estrutural quanto das redes neurais na interpretação e análise de dados complexos de forma mais simplificada.

## 1.2. Objetivo da pesquisa

Este trabalho tem como objetivo avançar na investigação sobre a viabilidade do uso de GNNs para a detecção e localização de vazamentos em RDAs. Para isso, propõe-se a utilização do algoritmo de GNN conhecido como Amostragem e Agregação de Grafos (*Graph Sample and Aggregation* - GraphSAGE) (Hamilton et al., 2017), consolidado na literatura, para realizar essa tarefa, aplicando três diferentes tipos de agregadores.

Além disso, será investigada a aplicação de uma matriz de ponderação baseada nas distâncias físicas entre os nós da rede, em substituição à tradicional matriz de adjacência binária, como forma de representar com maior fidelidade as relações estruturais da RDA e aprimorar a expressividade dos modelos GNN.

Para atingir o objetivo proposto, este estudo fundamenta-se na mesma estrutura de RDAs, bem como na metodologia de aplicação de GNNs utilizadas por Rolle (2024). Assim, este trabalho pode ser considerado uma continuação da pesquisa anterior, que também servirá como referência e *benchmark* para a avaliação dos resultados aqui obtidos.

A partir desse escopo, formulam-se as seguintes perguntas de pesquisa:

**P1:** A aplicação do algoritmo GraphSAGE melhora o desempenho da detecção e localização de vazamentos em RDAs modeladas como grafos, em comparação com algoritmos de GNNs utilizados anteriormente?

**P2:** O uso de uma matriz de ponderação baseada nas distâncias físicas entre os nós da rede, em vez da matriz de adjacência binária, melhora o desempenho do modelo GNN na tarefa de detecção e localização de vazamentos?

Para responder a essas questões, são formuladas as seguintes hipóteses:

- **$H_{01}$  (hipótese nula da **P1**):** O uso do algoritmo GraphSAGE, com diferentes agregadores, não apresenta melhora o desempenho de detecção e localização de vazamentos em RDAs, quando comparado aos algoritmos de GNNs utilizados anteriormente.
- **$H_{11}$  (hipótese alternativa da **P1**):** O uso do algoritmo GraphSAGE, com diferentes agregadores, apresenta melhora o desempenho de detecção e localização de vazamentos em RDAs, superando os algoritmos previamente utilizados como *benchmark*.
- **$H_{02}$  (hipótese nula da **P2**):** O uso de uma matriz de ponderação baseada nas distâncias físicas entre os nós da rede não melhora o desempenho do modelo GNN em comparação com a matriz de adjacência binária.
- **$H_{12}$  (hipótese alternativa da **P2**):** O uso de uma matriz de ponderação baseada nas distâncias físicas entre os nós da rede melhora o desempenho do modelo GNN em relação à matriz de adjacência binária.

Essas hipóteses serão avaliadas com base nas métricas de desempenho obtidas em diferentes cenários simulados. A análise dos resultados permitirá verificar não apenas se o GraphSAGE com diferentes agregadores apresenta superioridade em relação a outros algoritmos GNN, mas também se a substituição da matriz binária por uma matriz ponderada contribui de forma significativa para aprimorar a detecção e localização de vazamentos em RDAs.

### 1.3. Estrutura e conteúdo abordado

O presente trabalho é estruturado em 6 Capítulos, incluindo esta introdução e desconsiderando as referências bibliográficas.

No Capítulo 2, é realizada uma revisão bibliográfica, abordando a detecção e localização de vazamentos em RDAs, descrevendo os métodos tradicionais e a aplicação de aprendizado de máquina para essa tarefa, destacando também a utilização de GNNs neste contexto.

No Capítulo 3, é apresentada a fundamentação teórica, na qual se discute a modelagem de RDAs por meio do *software* EPANET, além de se abordar os fundamentos das GNNs, com ênfase na arquitetura do GraphSAGE e na introdução dos conceitos de matrizes de adjacência e de distâncias.

O Capítulo 4 descreve os materiais e métodos da pesquisa, detalhando o modelamento das RDAs no *software* EPANET, bem como a aquisição dos seus dados, através de simulações utilizando o EPANET-Matlab *Toolkit* (version 2.2). Nele também é apresentado o fluxograma da aplicação do algoritmo de GNN utilizado, descrevendo os hiperparâmetros adotados. Além disso, são apresentadas as métricas de avaliação empregadas para mensurar a performance do algoritmo.

No Capítulo 5, são expostos os resultados obtidos após o treinamento e validação do algoritmo nas RDAs hipotéticas utilizadas, que servem como base para discutir o potencial e as limitações das técnicas aqui implementadas.

Por fim, no Capítulo 6, são fornecidas as conclusões da pesquisa.

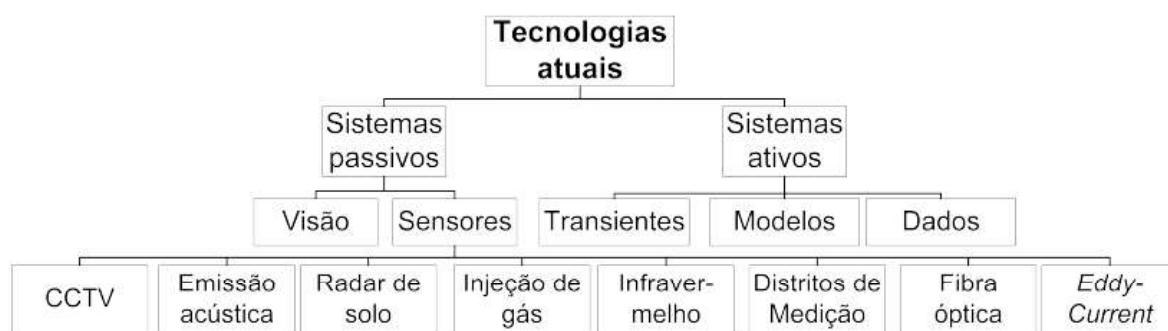
## 2. REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentada a revisão bibliográfica sobre as principais técnicas empregadas para a detecção e localização de vazamentos na rede de distribuição de água, considerando os métodos ativos e passivos, além da aplicação de algoritmos de aprendizado de máquina para a realização da tarefa.

### 2.1. Detecção e localização de vazamentos

Muitos fatores podem contribuir para vazamentos em RDAs, incluindo corrosão, desgaste, fissuras, movimentação de solo e interferências externas (Rojek & Studzinski, 2019). Scott e Barrufet (2003) sugerem uma classificação dos métodos de detecção de vazamentos em duas categorias principais: métodos baseados em *hardware* (externos), que utilizam sensores específicos, e métodos baseados em *software* (internos), que aproveitam sensores comuns da operação da tubulação, como medidores de pressão, vazão e temperatura. Embora esta distinção seja amplamente referenciada, ela foi expandida por abordagens mais recentes (Torres et al., 2020; Chan, Chin & Zhong, 2018). Conforme indicado na Figura 7, as técnicas passivas baseiam-se na inspeção visual ou monitoramento direto do sistema, enquanto as ativas recorrem à análise de sinais (como sinais acústicos, de vibração e de pressão) para detectar variações que possam indicar vazamentos.

Figura 1 - Panorama das técnicas empregadas na detecção de vazamentos em RDAs



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

#### 2.1.1. Métodos de detecção passiva

Segundo Chan, Chin e Zhong (2018) e Adedeji et al. (2017), os métodos passivos de detecção incluem:

- **Inspeção visual:** consiste na observação direta da superfície para identificar poças d'água ou crescimento anômalo de vegetação, que podem sinalizar a presença de um vazamento;
- **Sensoriamento:** são dispositivos portáteis que possibilitam a observação ou medição de variáveis relevantes. Exemplos incluem:
  - **Circuito Fechado de Televisão (CFTV):** sistema de captura de imagens para inspeção visual, que pode ser fixo ou montado em robôs;
  - **Sensores de emissão acústica:** captam o ruído característico de vazamentos, detectado por sensores instalados externamente à tubulação;
  - **Radar de solo:** dispositivo eletromagnético utilizado para identificar vazamentos em tubulações enterradas;
  - **Injeção de gás:** introdução de gás na tubulação, enquanto sensores externos monitoram o entorno para localizar vazamentos;
  - **Termografia por infravermelho:** técnica de detecção de vazamentos por meio da medição de radiação infravermelha em superfícies, útil para tubulações enterradas;
  - **Distritos de Medição e Controle:** subdivisões de um sistema de distribuição de água, onde são feitas medições de pressão e vazão;
  - **Sensores de fibra óptica:** monitoram alterações de temperatura em áreas próximas a vazamentos;
  - **Sensores de Eddy-Current:** empregados para detectar corrosão em tubulações de metal.

Os métodos apresentados são importantes e mostram bons resultados em condições favoráveis. Entretanto, eles se restringem à detecção de vazamentos que estão ativos no momento da inspeção, o que limita a possibilidade de monitoramento contínuo do sistema (Nkmeni, Mieyville & Tsafack, 2020). Algumas das limitações mais frequentes incluem o alto custo de certas tecnologias, como fibra óptica e CCTV; a necessidade de trabalho manual intensivo em métodos como inspeção visual, radar de solo e termografia por infravermelho; e a dependência significativa da experiência do operador, o que impacta diretamente a confiabilidade dos resultados obtidos (Chan, Chin & Zhong, 2018; Adedeji et al., 2017).

### 2.1.2. Métodos de detecção ativa

Os métodos de detecção ativa de vazamentos evitam a necessidade de inspeção visual ou por instrumentação manual, sendo classificados em três grandes abordagens, conforme Nkemeni, Mieyeville e Tsafack (2020) e Chan, Chin e Zhong (2018).

A primeira categoria é a abordagem baseada em transientes, que considera o vazamento como um fenômeno hidráulico identificável por meio de alterações de pressão no sistema. Modificações na estrutura física da tubulação (como vazamentos ou obstruções) causam reflexões de ondas nos sinais de pressão, as quais podem ser captadas e analisadas (Colombo, Lee & Karney, 2009). Nessas técnicas, transientes hidráulicos gerados por ações controladas (como a abertura de válvulas) ou pelo próprio vazamento produzem sinais que refletem atenuações características, comparáveis a sinais de referência, o que permite a detecção do vazamento (Medina, 2017). Modelagens em série de Fourier, por exemplo, são utilizadas para interpretar a magnitude do vazamento e sua localização, com base na taxa de amortecimento dos termos da série.

Dentre as técnicas para análise de transientes, destacam-se a transformada *wavelet*, análise harmônica, resposta em frequência com fluxo oscilatório e transiente inverso (Medina, 2017; Abdulshaheed, Mustapha & Ghavamian, 2017). Uma revisão abrangente de Ayati, Haghighi e Lee (2019) compila diversas abordagens de transientes, observando que a maioria dos estudos ocorre em ambientes controlados e que apenas cerca de 4% dos trabalhos até 2018 relatam testes em campo, evidenciando a dificuldade de aplicação prática. As limitações incluem a alta resolução temporal exigida para captar transitórios (em torno de 1 kHz), o que demanda sensores de alta precisão e encarece a infraestrutura (Medina, 2017).

A segunda categoria, a abordagem baseada em modelos hidráulicos, utiliza representações matemáticas simplificadas dos componentes do sistema (reservatórios, bombas e tubulações) integradas a equações de conservação de massa e energia para replicar o comportamento da rede (Medina, 2017). Com o avanço da informática, esses modelos foram aprimorados para processar grandes volumes de dados rapidamente, substituindo cálculos tradicionais (Silva, 2014). Neste contexto, vazamentos são representados como fluxos dependentes da pressão em locais específicos da rede. Contudo, limitações incluem a variabilidade temporal das condições da tubulação e o desafio de modelar demandas de consumo de maneira acurada, além de incertezas quanto à rugosidade e vazamentos já existentes (Chan, Chin & Zhong, 2018). Estimações baseadas em filtros de Kalman, observadores de estados e métodos de impedância, oferecem alternativas para obter parâmetros em tempo real (Nkemeni, Mieyeville & Tsafack, 2020).

A terceira categoria, segundo Chan, Chin e Zhong (2018), é a abordagem baseada em dados (*data-driven*). Este método utiliza dados de monitoramento contínuo sem manobras específicas, dispensando a complexidade dos modelos hidráulicos detalhados (Medina, 2017). Utilizando conceitos de Internet das Coisas (*Internet of Things – IoT*), sensores de pressão e vazão são distribuídos ao longo da rede, fornecendo grande volume de dados, que podem ser analisados por algoritmos para detectar anomalias e localizar vazamentos (Liang et al., 2020; Zhou et al., 2019). Essas metodologias de monitoramento contínuo minimizam interrupções e facilitam a identificação precisa de vazamentos em redes urbanas, por meio de modelos que aprendem os padrões normais de operação (Romano, Kapelan & Savić, 2014; Loureiro et al., 2016).

### **2.1.3. Aprendizado de máquina para detecção de vazamentos**

O uso de Aprendizado de Máquina (*Machine Learning - ML*) na detecção de vazamentos em RDAs tem avançado consideravelmente nos últimos anos. A crescente digitalização e a infraestrutura de *IoT*, facilitam a obtenção de grandes volumes de dados, permitindo que os métodos de ML identifiquem padrões complexos de consumo e pressão para detectar vazamentos com maior agilidade, reduzindo desperdícios e minimizando custos operacionais. Abaixo, serão apresentados alguns trabalhos, da literatura recente, que focam na tarefa da detecção de vazamentos em RDAs utilizando técnicas de ML. Este levantamento não visa realizar uma revisão sistemática da literatura, mas sim destacar os avanços recentes e exemplos de aplicação que demonstrem a viabilidade das técnicas de ML na detecção de vazamentos em RDAs.

Oliveira et al. (2018) aplicou técnicas de ML para detectar vazamentos em uma tubulação de 5 km destinada ao transporte de rejeitos em uma usina de mineração no Brasil. Utilizando medições de pressão, o estudo empregou regressão linear para desenvolver um modelo preditivo da energia ao longo da tubulação, permitindo a identificação de comportamentos anômalos indicativos de vazamentos.

Em Javadiha et al. (2019), foi proposta uma abordagem para detecção de vazamentos em RDAs baseada no cálculo de resíduos de pressão. Por meio de interpolação, foi estimada a pressão em todos os nós da rede a partir dos pontos de medição disponíveis, criando assim um mapa de resíduos de pressão. Esses mapas (imagens) foram utilizados para treinar uma Rede Neural Convolucional (*Convolutional Neural Network - CNN*) capaz de identificar em qual nó há maior probabilidade de conter vazamento. Em um estudo de caso utilizando simulação de uma RDA real no EPANET, com dados de vazão e pressão coletados a cada 10 minutos, a

metodologia alcançou acurácia de 94%, dependendo da quantidade de sensores alocados na rede.

No trabalho de Sun et al. (2020), foi proposto uma metodologia para localização de vazamentos baseada em dados de pressão, destacando que sensores de pressão são mais econômicos e fáceis de instalar em comparação aos de vazão. O estudo utilizou duas abordagens, sendo elas a Análise Discriminante Linear (*Linear Discriminant Analise* - LDA) e uma Rede Neural Artificial (RNA) com uma camada oculta. Para testar a metodologia, foi realizada uma simulação da RDA da cidade de Hanói. Os resultados indicaram que tanto a quantidade quanto o posicionamento dos sensores influenciam significativamente a precisão da detecção. Diferente de estudos anteriores, os autores optaram por utilizar o sinal de pressão bruto, descartando o uso de residuais devido à ineficácia da interpolação em representar as condições reais da RDA. A acurácia obtida variou entre 55% e 75%, dependendo do número de sensores empregados.

Em Pérez-Pérez et al. (2021), os autores estudaram a problemática da localização de vazamentos utilizando uma abordagem baseada na combinação de RNAs para estimar o fator de fricção da tubulação. Três RNAs foram treinadas, sendo duas para estimar o fator de fricção e uma para localizar o vazamento. Os testes ocorreram em uma planta piloto de laboratório, complementados por um modelo computacional dessa planta, para simular cenários adicionais de vazamento. Foram realizadas medições de vazão e pressão em dois pontos, com uma taxa de amostragem de 1 kHz. O estudo encontrou um erro médio percentual de 0,47% na localização do vazamento, mas destaca que fatores como ruídos, maiores distâncias entre sensores e problemas de comunicação podem impactar a precisão da técnica em sistemas reais.

No estudo de Fan et al. (2021), foram desenvolvidos modelos de ML para a detecção de vazamentos em redes de abastecimento de água, utilizando principalmente RNAs e *Autoencoders* (AE). Os dados de pressão da água, simulados com o *software* EPANET, foram empregados para treinar esses modelos em condições com e sem vazamento. A ANN foi utilizada para classificar dados balanceados, enquanto o AE, treinado em dados não balanceados e não rotulados, identificou anomalias com base no erro de reconstrução. Uma estratégia inovadora de múltiplas tentativas independentes foi implementada para aumentar a precisão da detecção e reduzir alarmes falsos. Os resultados indicaram alta precisão na detecção de vazamentos, especialmente dentro das áreas monitoradas, sugerindo que o modelo pode ser utilizado para monitoramento em tempo real em redes de abastecimento de água, integrando tecnologias *IoT* para melhorar a confiabilidade dos dados obtidos por sensores.

Em Lučin, I. et al. (2021), um algoritmo de classificação baseado em Floresta Aleatória (*Random Forest*) foi empregado para localizar vazamentos RDAs de diferentes tamanhos, utilizando sensores distribuídos de forma esparsa. Foram simulados cenários de vazamentos com variáveis como localização e coeficiente do emissor, levando em conta a variação da demanda ao longo do dia. A metodologia incluiu a construção de modelos preditivos que avaliaram a precisão do classificador sob diferentes condições de vazamento e demanda. Os resultados indicaram que o modelo apresentava maior precisão em vazamentos maiores e menor variação na demanda base. Utilizando o *software* EPANET para simular dados de pressão nos nós da rede, o modelo alcançou até 89% de precisão em redes menores, ao considerar os cinco nós mais prováveis de conter vazamento. Essa abordagem demonstrou potencial para aplicação em RDAs reais, uma vez que permite gerar uma quantidade significativa de dados sintéticos para compensar a escassez de dados reais sobre eventos de vazamento.

#### **2.1.4. Aplicação baseada em grafos**

A aplicação de métodos baseados em grafos para detecção e localização de vazamentos em RDAs tem a capacidade de aproveitar a estrutura natural dessas redes, representadas como grafos, onde cada ponto (nó) e conexão (aresta) entre tubulações e sensores são mapeados. Esse modelo permite uma análise detalhada das interdependências estruturais e hidráulicas da rede, possibilitando a identificação de padrões de pressão e vazão que podem sinalizar vazamentos.

Um exemplo de análise baseada em grafos foi proposto no estudo conduzido por Candelieri et al. (2014), onde o intuito foi o aprimoramento do gerenciamento de vazamentos em RDAs através do método de clusterização, resultando na localização dos pontos de vazamento. A metodologia foi aplicada a dados de simulação hidráulica gerados pelo *software* EPANET, revelando a superioridade da clusterização baseada em grafo em relação às técnicas convencionais de ML.

Outro exemplo se dá em Zanfei et al. (2022), onde o estudo realizado propôs um novo *framework* de gestão para RDAs, utilizando modelos de Redes Convolucionais de Grafos (*Graph Convolutional Network – GCN*) para detectar rompimentos nas tubulações. Os métodos desenvolvidos dependeram de uma representação em grafo da RDA, utilizando medidas de pressão e vazão. Dois modelos de GCN foram projetados com base nessa representação, para identificar rompimentos. Adicionalmente, dois modelos de RNAs foram utilizados como referência para comparação. A metodologia foi testada em uma rede de utilidade de água, destacando o potencial das GCNs na detecção de anomalias em RDAs.

Por fim, em Rolle et al. (2024), foi proposto uma metodologia para localização de vazamentos em RDAs utilizando GNNs. A pesquisa utilizou uma RDA simulada para o treinamento e a validação do modelo, implementando uma técnica de regressão linear para estimar dados de nós não monitorados, o que reduziu o número de sensores e os custos de implementação. A abordagem consistiu em representar a rede de água como um grafo, com nós correspondentes a pontos de consumo e junções de tubulações. Em três cenários de simulação, o modelo GNN foi treinado e testado com (1) dados completos, (2) dados sem informações das junções e (3) dados estimados nas junções por regressão linear. Os resultados mostraram que a inclusão da regressão linear melhorou a precisão do modelo em até 18% em comparação com o cenário sem dados das junções. Mesmo com uma redução de 37% no número de sensores, a metodologia alcançou uma acurácia de até 80%, indicando sua viabilidade para aplicações práticas de monitoramento e localização de vazamentos.

A revisão apresentada evidencia a diversidade de abordagens utilizadas na detecção e localização de vazamentos em Redes de Distribuição de Água (RDAs), desde métodos passivos e ativos até soluções baseadas em aprendizado de máquina. Embora as técnicas convencionais ainda tenham aplicação relevante, suas limitações em termos de custo, escalabilidade e sensibilidade reforçam a necessidade de métodos mais robustos e automatizados. Nesse contexto, os modelos de aprendizado de máquina demonstram crescente eficácia, especialmente quando aliados à simulação computacional e ao uso de sensores. Dentre essas técnicas, destaca-se a aplicação de Redes Neurais de Grafos (GNNs), que se mostram particularmente promissoras por sua capacidade de explorar a topologia da rede e extrair relações estruturais complexas. A literatura recente confirma que o uso de GNNs pode superar abordagens tradicionais ao incorporar a estrutura relacional das RDAs no processo de aprendizado.

#### **2.1.5. Uso do GraphSAGE para Detecção de Anomalias**

Nos últimos anos, algoritmos de aprendizado em grafos, em especial o GraphSAGE, têm ganhado destaque como ferramentas eficazes para a detecção de anomalias em diferentes domínios. O GraphSAGE é um modelo de rede neural de grafos do tipo convolucional, caracterizado por sua capacidade indutiva: ele aprende uma função de agregação que permite gerar embeddings para nós nunca vistos anteriormente, utilizando as características dos nós vizinhos e suas conexões. Essa característica é particularmente vantajosa em cenários dinâmicos, onde novas informações ou nós podem surgir continuamente, como ocorre em sistemas complexos.

A utilização do GraphSAGE para detecção de anomalias tem sido explorada em diferentes áreas. No trabalho de Abou Rida et al. (2021), por exemplo, foi proposto um sistema para identificar ataques em redes de IoT representadas como grafos. A metodologia envolveu a construção de um grafo a partir de eventos de rede e a aplicação de técnicas de análise de grafos, métodos de embeddings e redes neurais de grafos. O GraphSAGE apresentou resultados superiores na identificação de nós anômalos quando comparado a técnicas tradicionais, evidenciando sua robustez em ambientes dinâmicos e complexos.

Chen et al. (2023) aplicaram o GraphSAGE em sistemas de energia, incorporando informações temporais e estruturais dos vizinhos para detectar comportamentos anômalos na rede elétrica. Ao gerar embeddings capazes de capturar simultaneamente padrões temporais e estruturais, o modelo conseguiu identificar eventos atípicos que não seriam detectados por métodos convencionais. Essa abordagem mostrou como o GraphSAGE pode integrar múltiplas fontes de informação para detectar falhas ou eventos anômalos em sistemas críticos.

Mais recentemente, Marfo et al. (2024) exploraram uma arquitetura híbrida combinando GraphSAGE e Graph Attention Networks (GAT) para detecção de anomalias em tráfego de rede. Nessa abordagem, o GraphSAGE foi responsável pela construção inicial dos embeddings dos nós, a partir das conexões locais, enquanto o GAT refinou a atenção sobre as conexões mais relevantes. O modelo alcançou desempenho significativamente superior aos modelos individuais, com acurácia próxima de 98%, reforçando o potencial do GraphSAGE em aplicações de segurança e monitoramento de sistemas.

A partir dessas aplicações, é possível notar o potencial do GraphSAGE para lidar com problemas de detecção de padrões atípicos em sistemas complexos. No contexto das Redes de Distribuição de Água (RDAs), a estrutura da rede pode ser naturalmente representada como um grafo, no qual cada nó corresponde a um ponto da rede (como sensores, junções ou reservatórios) e cada aresta representa a conexão hidráulica entre esses elementos. Assim, técnicas baseadas no GraphSAGE podem ser adaptadas para detectar e localizar vazamentos, tratando os vazamentos como anomalias no comportamento hidráulico. A capacidade do GraphSAGE de aprender representações de nós considerando tanto atributos locais (como pressão e vazão) quanto a topologia da rede permite identificar mudanças sutis que indicam vazamentos, mesmo em cenários com dados incompletos ou dinâmicos. Desse modo, este trabalho avança nesse campo ao investigar o impacto da utilização do algoritmo GraphSAGE, com diferentes estratégias de agregação e ponderações baseadas em distância física, visando ampliar a acurácia na detecção e localização de vazamentos, contribuindo com uma abordagem mais realista e eficaz para o monitoramento de perdas em sistemas hidráulicos.

### 3. FUNDAMENTAÇÃO TEÓRICA

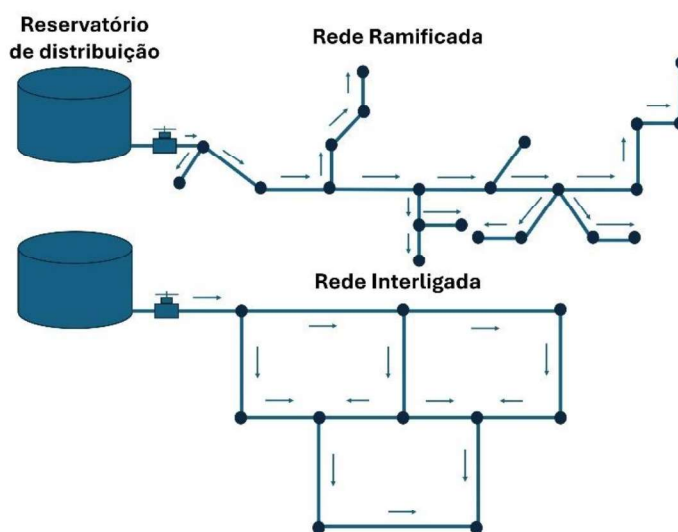
Neste capítulo será apresentada a fundamentação teórica de três conceitos essenciais para este estudo. O primeiro aborda a modelagem de RDAs, que possibilita a simulação computacional e a obtenção de dados para o treinamento dos algoritmos de ML. O segundo trata das GNNs, com ênfase na técnica GraphSAGE, que é aplicada neste trabalho. Por fim, o terceiro conceito discute a representação de grafos por meio de matrizes de adjacência e matrizes ponderadas.

#### 3.1. Modelagem de redes de distribuição de água

Os modelos hidráulicos de distribuição de água, desenvolvidos em *softwares* de simulação, são ferramentas amplamente utilizadas para operação e gestão de recursos hídricos, fornecendo suporte essencial aos operadores na tomada de decisão em diferentes cenários de anormalidade (Tejaswini; Silo; Ashwini, 2020). Esses modelos representam a RDA de forma simplificada, integrando seus componentes (tubulações, reservatórios, bombas, entre outros) a equações que descrevem o comportamento do sistema hidráulico, permitindo o cálculo de variáveis de estado, como pressão, vazão e perda de carga (Medina, 2017).

As RDAs podem ser interligadas ou ramificadas, como ilustrado na Figura 2. Um sistema interligado oferece múltiplos caminhos para que a água alcance diferentes pontos da rede, aumentando a confiabilidade, mas com custos de implementação mais elevados. Em contraste, um sistema ramificado, onde a água flui dos reservatórios para os pontos de consumo por um único caminho, o custo de instalação é reduzido, porém há também a diminuição de sua confiabilidade, já que uma falha em determinado ponto pode afetar um número maior de clientes (Walski et al., 2003).

Figura 2 – Exemplo de redes de distribuição



Fonte: Autoria própria.

O EPANET é um dos *softwares* mais utilizados para modelar RDAs, sendo concebido para ser uma ferramenta de apoio à análise de sistemas de distribuição, melhorando o conhecimento sobre o transporte e o destino dos constituintes da água para o consumo humano (Rossman et al., 1994). Trata-se de uma ferramenta de domínio público desenvolvida pela Agência de Proteção Ambiental dos Estados Unidos EPA (*US Environmental Protection Agency*), que é responsável pela conservação dos recursos naturais no país (Tejaswini; Silo; Ashwini, 2020). O aplicativo envolve aspectos tanto de cálculo hidráulico (como pressão, vazão e perdas de carga), quanto de análise de qualidade da água e contaminação. Por essas capacidades, o EPANET se tornou uma ferramenta padrão para a indústria, onde permite simular diversos cenários de possíveis falha e avaliar seus efeitos, e também para a academia, onde possibilita a geração de dados para a avaliação de novos algoritmos e métodos de monitoramento de RDAs (Eliades et al., 2016).

Contudo, como o EPANET é um aplicativo independente e autossuficiente, ele não permite programar rotinas de simulação, nem obter dados de forma automatizada, sendo necessária a sua execução manual. Para contornar essa limitação, foi desenvolvido o EPANET-MATLAB Toolkit, que possibilita o acesso a todas as funções e bibliotecas do EPANET a partir do MATLAB, um *software* de processamento matemático (Eliades et al., 2016). O toolkit permite que o usuário importe mapas de RDAs criados na interface gráfica do EPANET, e, no ambiente do MATLAB, altere a rede por meio de código, o que inclui habilitar ou desabilitar nós de vazamento, modificar padrões de demanda, ajustar curvas de bombas e, ainda, armazenar de forma padronizada os dados resultantes de cada simulação realizada.

### 3.1.1. Componentes de um modelo

Uma RDA consiste em vários trechos de canos interligados, cada qual com seu comprimento, diâmetro e rugosidade específicos, definidos pelo material dos tubos. Esses trechos podem conter bombas e acessórios, como cotovelos e válvulas (Bermúdez et al., 2018). As bombas são empregadas para superar elevações e perdas de pressão causadas pela fricção entre a água e a tubulação, enquanto as válvulas são utilizadas para controlar o fluxo e regular a pressão (Perez et al., 2014).

Os nós na rede podem ser classificados em dois tipos, sendo em um nó simples, representando um ponto de consumo dentro da rede, ou em um nó de junção, que representa o encontro de dois ou mais trechos de tubulação, podendo também ser um ponto de consumo com fluxo de entrada ou saída de água do sistema (Bermúdez et al., 2018).

Na modelagem de sistemas de monitoramento de vazão, nós adicionais podem ser inseridos para representar pontos de medição, bem como vazamentos, utilizando o coeficiente de emissor, conforme descrito posteriormente (Seção 3.1.7).

A rede conta ainda com dois tipos de reservatórios: os Reservatórios de Nível Fixo (RNF), com capacidade ilimitada e carga hidráulica constante, que representam origens ou sumidouros de água, e os Reservatórios de Nível Variável (RNV), com capacidade limitada e volume variável ao longo da simulação, modelando reservatórios como caixas d'água.

### 3.1.2. Lei da conservação de massa

O fluxo de fluidos em regime permanente obedece às leis da Conservação de Massa e da Conservação de Energia. A Lei da Conservação de Massa estabelece que a massa de fluido que entra em uma tubulação é igual à massa que sai dela (Walski et al., 2003; Perez et al., 2014). Na modelagem de RDAs, todo o fluxo de saída de fluido é implementado nos nós. Esta lei é expressa conforme Equação 1.

$$\sum_{\text{canos}} Q_i - U = 0 \quad (1)$$

onde  $Q_i$  representa o fluxo de entrada do nó a partir do  $i$  –ésimo cano, e  $U$  corresponde ao consumo de água pelo nó (WALSKI et al., 2003).

Em simulações de longo período, a água pode ser armazenada ou consumida em reservatórios distribuídos pela rede. Assim, um novo termo é adicionado à equação de Conservação de Massa, ilustrado na Equação 2.

$$\sum_{\text{canos}} Q_i - U - \frac{dS}{dT} = 0 \quad (2)$$

onde  $\frac{dS}{dT}$  é a variação de volume de água nos reservatórios.

### 3.1.3. Lei da conservação de energia

A Lei da Conservação de Energia estabelece que a diferença de energia entre dois pontos é igual à energia adicionada ao fluxo pelos componentes entre esses pontos, subtraídas as perdas de carga (seja por fricção ou perdas menores) (Perez et al., 2014). Para fins de análise hidráulica, a Equação 3 expressa esses termos em forma de carga.

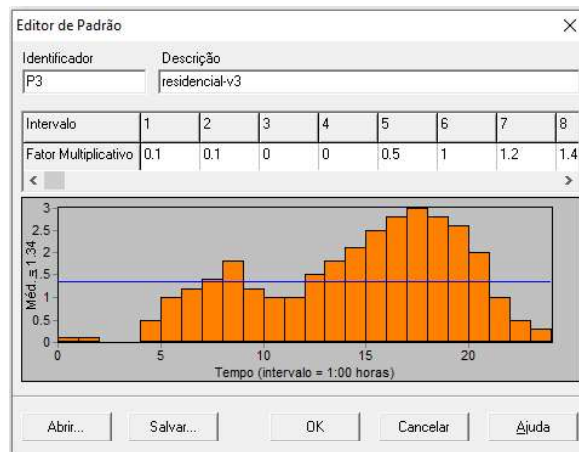
$$1 + \frac{P_1}{\gamma} + \frac{V_1^2}{2g} + \sum h_p = Z_2 + \frac{P_2}{\gamma} + \frac{V_2^2}{2g} + \sum h_L + \sum h_m \quad (3)$$

onde  $Z$  representa a elevação,  $P$  a pressão,  $\gamma$  a massa específica do fluido,  $V$  a velocidade de escoamento,  $g$  a aceleração gravitacional,  $h_p$  a carga adicionada por bombas,  $h_L$  a perda de carga nos canos e  $h_m$  as perdas de carga menores (Walski et al., 2003).

### 3.1.4. Consumo de água

Em cada nó  $i$  de consumo é atribuído um valor de consumo-base, representando o consumo médio ou nominal de água da categoria predominante de uso, expresso em unidades de vazão (Rossman et al., 1994). Esse valor pode ser positivo (indicando consumo de água), nulo (indicando ausência de consumo) ou negativo (indicando uma fonte externa de vazão no nó). Além disso, cada nó possui um padrão temporal de consumo, composto por múltiplos fatores multiplicativos que representam as variações de consumo ao longo da simulação. Por exemplo, em uma simulação de 24 horas, o padrão de consumo pode ser configurado para intervalos horários, refletindo as flutuações na demanda de água ao longo de um dia. Dessa forma, uma única RDA pode conter diversos padrões de consumo, ilustrando perfis distintos de consumidores, como residencial, industrial ou rural, entre outros. A Figura 3 apresenta um exemplo fictício de um padrão de consumo.

Figura 3 – Padrão de consumo em uma RDA para o intervalo 24 horas



Fonte: Autoria própria.

Assim, o consumo de água em um nó, em determinado instante, é calculado pelo produto entre o consumo-base atribuído ao nó e o fator multiplicativo do padrão de consumo para aquele momento específico (Rossman et al., 1994). Dessa forma, a demanda de cada nó é definida pela conforme Equação 4.

$$d_i = bd_i p_{a,i}(k) \quad (4)$$

onde  $bd_i$  é o valor de consumo base,  $p_{a,i}(k)$  é o valor do padrão  $a$  associado ao nó  $i$  no instante de tempo  $k$  (Walski et al., 2003).

### 3.1.5. Perdas de carga

As perdas de carga nos canos, decorrentes da fricção, podem ser representadas por diferentes modelos de equações. Uma das mais representações mais comuns pode ser vista na Equação 5.

$$h_L = Kq^T \quad (5)$$

onde  $K$  é um coeficiente da tubulação que varia conforme o diâmetro, comprimento e material do cano,  $q$  representa a vazão, e  $r$  é um expoente próximo de dois (Perez et al., 2014; Walski et al., 2003). O coeficiente  $K$  pode ser calculado usando três métodos distintos, sendo as equações de Darcy-Weisbach, Hazen-Williams ou Chezy-Manning, todos disponíveis para aplicação no *software* EPANET (Rossman et al., 1994).

A simulação hidráulica calcula as cargas nos nós e nas tubulações para um conjunto específico de níveis de reservatório e demanda de consumo de água. A solução em um determinado instante requer a resolução simultânea do sistema de equações não lineares de conservação de massa e energia aplicadas a todos os nós e trechos da rede (Perez et al., 2014).

No EPANET, cada trecho da tubulação tem dados como comprimento, diâmetro e um parâmetro relacionado às perdas de carga. No método de Hazen-Williams (padrão do EPANET), esse parâmetro é chamado de “Fator C”, que indica a capacidade de transporte da tubulação, sendo quanto maior o Fator C, mais facilmente a água se desloca pelo tubo. Walski et al. (2003) disponibiliza uma lista com os valores de Fator C para diversos materiais de tubulação.

Além das perdas por fricção, ocorrem perdas menores devido aos elementos de controle de fluxo (como válvulas) e ao trajeto da tubulação (curvas, junções em “T”, entre outros), conhecidas como perdas de carga singulares. A passagem de água por esses elementos aumenta a turbulência e resulta em perda de carga adicional. Em Rossman et al. (1994), são fornecidos os coeficientes de perda de carga singular para diversos elementos.

### **3.1.6. Controles**

Segundo Rossman et al. (1994), os controles são instruções que definem o modo de operação da rede ao longo do tempo, permitindo a alteração do estado de trechos ou bombas da RDA. Os controles podem ser simples ou programados.

Controles simples modificam o estado ou propriedades de um trecho com base em parâmetros específicos, como altura de água em um Reservatório de Nível Variável (RNV), pressão em um nó, tempo de simulação ou hora do dia. Já os controles programados permitem que o estado e as propriedades de um trecho dependam da combinação de um conjunto de condições ocorrentes na rede, após o cálculo das condições hidráulicas iniciais (Rossman et al., 1994).

### **3.1.7. Modelagem de vazamentos**

As perdas de água, comumente conhecidas como vazamentos, podem ser categorizadas em duas principais modalidades, sendo rupturas súbitas (“*burst*”) e aberturas ou fissuras nas tubulações. As rupturas súbitas são geralmente mais fáceis de identificar, pois resultam em mudanças significativas na dinâmica da RDA e geram danos visíveis, como acúmulos de água e deslocamento do solo. Em contraste, as fissuras ou buracos nas tubulações costumam ser mais desafiadores de detectar, demandando o uso de técnicas mais avançadas para sua localização.

Nas simulações de modelos matemáticos de RDAs, os nós são utilizados como sendo os pontos de vazamentos. Diferentemente de um consumo constante, os vazamentos são representados de forma mais realista por meio de um coeficiente de emissão  $C_e$ , que indica o tamanho do vazamento. Esse coeficiente gera um fluxo de vazamento proporcional à pressão no nó, conforme representado pela Equação 6.

$$q_l = C_e \cdot P^\gamma \quad (6)$$

onde  $q_l$  representa a vazão no orifício,  $P$  indica a pressão no nó e  $\gamma$  é um expoente associado à geometria do orifício. Este expoente pode atingir valores em torno de 2,3 para tubulações de metal corroído, enquanto para tubulações plásticas, ele varia entre 0,41 e 1,85. Em situações em que não há dados experimentais disponíveis (como em estudos inteiramente computacionais), o valor de 0,5 é utilizado para modelar orifícios circulares (Perez et al., 2014).

### 3.2. Redes neurais de grafo

As redes neurais de grafos (GNNs) (Scarselli et al., 2009) são uma classe de modelos de aprendizado de máquina que operam em dados estruturados em forma de grafos, onde a representação de relacionamentos entre os dados (nós e arestas) desempenha um papel central no processamento de informações. Elas têm sido amplamente utilizadas em uma variedade de aplicações, incluindo modelagem de sistemas físicos, previsão de interações moleculares, análise de redes sociais e processamento de linguagem natural (Wu et al., 2021). As GNNs são projetadas para capturar informações contextuais e relacionais entre os nós e as arestas do grafo, permitindo uma generalização eficaz para dados não vistos durante o treinamento (Zhou et al., 2020).

A arquitetura de uma GNN é composta por várias camadas que operam localmente em cada nó do grafo e agregam informações dos nós adjacentes, permitindo que o modelo aprenda representações de alta qualidade que refletem tanto as características de cada nó quanto as conexões entre eles (Battaglia et al., 2018). Em cada camada, as GNNs usam uma função de agregação que combina dados de nós vizinhos para atualizar o estado de cada nó, o que cria uma estrutura hierárquica de representação. Essa abordagem, chamada de *message passing*, permite que informações sobre o contexto local e as relações do grafo sejam compartilhadas e ampliadas progressivamente, o que resulta em uma representação mais robusta do grafo global (Gilmer et al., 2017).

Diversas variantes de GNNs foram desenvolvidas para atender a diferentes desafios de modelagem, incluindo as *Graph Convolutional Networks* (GCNs), *Graph Attention Networks* (GATs) e *Graph Recurrent Neural Networks* (GRNNs). As GCNs utilizam convoluções para simplificar a agregação de informações dos nós vizinhos, uma abordagem que tem mostrado alta eficácia em problemas como classificação e previsão de grafos (Kipf & Welling, 2017). As GATs, por outro lado, empregam mecanismos de atenção para ponderar a relevância dos nós vizinhos, o que permite uma agregação mais refinada e adaptável aos padrões complexos nos dados (Veličković et al., 2018).

O aprendizado em GNNs também pode ser supervisionado, não supervisionado ou semi-supervisionado, e cada uma dessas abordagens é usada conforme as características dos dados disponíveis e o objetivo do modelo. Em problemas supervisionados, como classificação de nós ou arestas, o modelo é treinado para prever rótulos com base nas características dos nós e na estrutura do grafo. Já em tarefas não supervisionadas, como aprendizado de representação, as GNNs são usadas para gerar embeddings de nós que capturam as propriedades estruturais da rede e podem ser aplicadas em tarefas *downstream* (Hamilton et al., 2017).

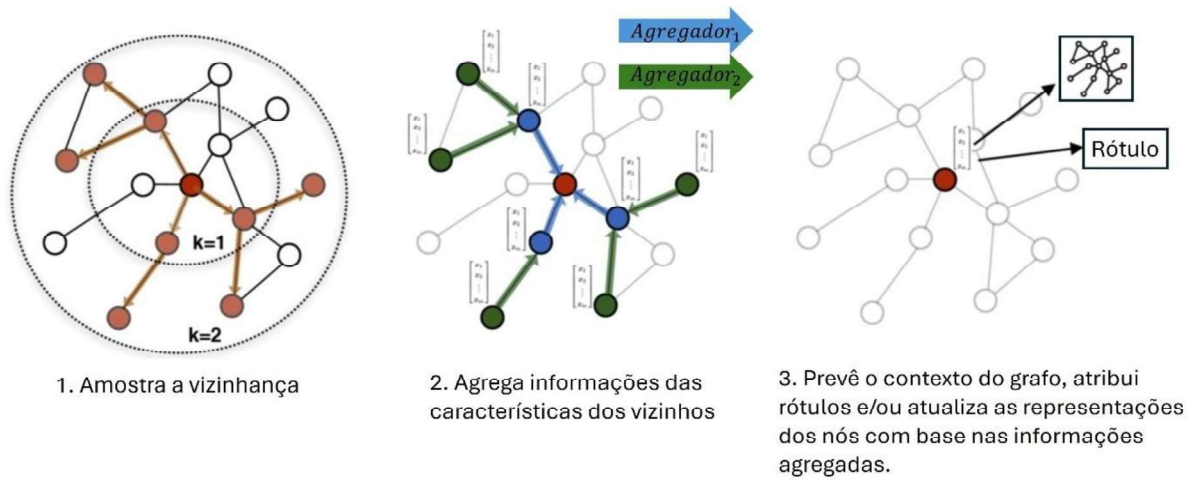
A capacidade das GNNs de generalizar para novos dados é um dos principais fatores de sua popularidade, e pesquisas recentes indicam avanços contínuos na eficiência e escalabilidade desses modelos, abrindo novas possibilidades para análises em grafos complexos (Xu et al., 2018). Isso é particularmente importante, pois a generalização efetiva dos modelos é essencial para lidar com os dados complexos e heterogêneos que são comuns em muitas aplicações práticas, permitindo uma aplicação robusta em áreas como biologia computacional, redes de informação, e ciências sociais.

### 3.2.1. Arquitetura e funcionamento do GraphSAGE

O algoritmo de Amostragem e Agregação de Grafos (*Graph Sample and Aggregation* - GraphSAGE) é uma técnica de aprendizado supervisionado projetada para gerar *embeddings* de nós em redes (grafos), permitindo a generalização para nós não vistos durante o treinamento. Proposto por Hamilton et al. (2017), o algoritmo introduz uma abordagem de amostragem e agregação de vizinhos locais para aprendizado eficiente em grafos grandes e dinâmicos. Ao contrário de outros métodos de aprendizado em grafos, que exigem que todo o grafo seja processado de uma vez, o GraphSAGE amostra um conjunto de vizinhos para cada nó e utiliza funções agregadoras, que combinam as características dessa vizinhança para gerar uma representação de baixa dimensionalidade. Essa representação é então utilizada para formar o *embedding* final do nó. A Figura 4 ilustra esse processo de maneira simplificada. Esse método

é amplamente utilizado em tarefas como classificação de nós e predição de arestas em redes complexas. O restante do conteúdo desta seção é baseado no estudo realizado por Hamilton et al. (2017).

Figura 4 - Ilustração visual da abordagem de amostragem e agregação do GraphSAGE



Fonte: Adaptado de Hamilton et al. (2017).

### 3.2.1.1. Algoritmos de geração de *embeddings*

O algoritmo de geração de embeddings, ou propagação direta (Algoritmo 1), parte do pressuposto de que o modelo já foi treinado e seus parâmetros estão fixos. Considera-se que foram aprendidos os parâmetros de  $K$  funções de agregação (representadas por  $AGGREGATE_k$ ,  $\forall k \in \{1, \dots, K\}$ ), responsáveis por combinar informações dos vizinhos de cada nó, além de um conjunto de matrizes de pesos  $W^k$ ,  $\forall k \in \{1, \dots, K\}$ , utilizado para a propagação de informações entre diferentes camadas do modelo, também chamadas de “níveis de busca”.

---

**Algoritmo 1:** Algoritmo de geração de embeddings do GraphSAGE (propagação para frente)
 

---

Entrada: Grafo  $G(V, E)$ ; características de entrada  $\{x_v, \forall v \in V\}$ ; profundidade  $K$ ; matrizes de pesos

$W_k, \forall k \in \{1, \dots, K\}$ ; não-linearidade  $\sigma$ ; funções de agregação diferenciáveis

$AGGREGATE_k, \forall k \in \{1, \dots, K\}$ ; função de vizinhança  $N: v \rightarrow 2^V$

Saída: Representações vetoriais  $z_v$  para todos  $v \in V$

```

1  $h_v^0 \leftarrow x_v, \forall v \in V;$ 
2 for  $k = 1 \dots K$  do
3   for  $v \in V$  do
4      $h_{N(v)}^k \leftarrow AGGREGATE_k(\{h_u^{k-1}, \forall u \in N(v)\});$ 
5      $h_v^k \leftarrow \sigma(W^k \cdot CONCAT(h_v^{k-1}, h_{N(v)}^k))$ 
6   end
7    $h_v^k \leftarrow h_v^k / \|h_v^k\|_2, \forall v \in V$ 
8 end
9  $z_v \leftarrow h_v^K, \forall v \in V$ 

```

---

O Algoritmo 1 descreve o processo de geração de *embeddings* a partir de um grafo completo,  $G = (V, E)$ , considerando que as características de todos os nós,  $x_v, \forall v \in V$ , são fornecidas como entrada. A cada iteração, também chamada de profundidade de busca, os nós reúnem informações de seus vizinhos imediatos e, conforme esse processo se repete, passam a incorporar dados de regiões mais distantes do grafo, ampliando gradualmente sua compreensão da estrutura global. Em cada passo  $k$ , onde  $h^k$  representa a representação do nó no nível atual, cada nó  $v \in V$  agrega as representações dos nós da sua vizinhança imediata,  $\{h_u^{k-1}, \forall u \in N(v)\}$ , em um vetor único  $h_{N(v)}^{k-1}$ . Essa etapa de agregação depende diretamente das representações geradas na iteração anterior (ou seja,  $k - 1$ ), sendo que, no caso inicial ( $k = 0$ ), as representações são definidas pelas características de entrada dos nós. Em seguida, o GraphSAGE concatena a representação atual do nó,  $h_v^{k-1}$ , com o vetor de vizinhança agregado,  $h_{N(v)}^{k-1}$ , e o resultado é processado por uma camada totalmente conectada, que utiliza uma função de ativação não linear  $\sigma$ . Essa transformação gera as representações que serão utilizadas na iteração seguinte (ou seja,  $h_v^k, \forall v \in V$ ), até alcançar a profundidade  $K$ , onde as representações finais são denotadas por  $z_v \equiv h_v^K, \forall v \in V$ . A agregação das representações dos vizinhos pode ser realizada por meio de diversas arquiteturas de agregação (representadas por  $AGGREGATE$  no Algoritmo 1), as quais serão detalhadas na Seção 3.2.1.3.

A vizinhança é definida como sendo uma amostragem uniforme de um conjunto de vizinhos de tamanho fixo, que, em vez de se utilizar conjuntos completos de vizinhança no Algoritmo 1, visa manter a carga computacional de cada lote constante. Assim,  $N(v)$  é definido como uma amostra uniforme de tamanho fixo do conjunto  $\{u \in V: (u, v) \in E\}$ , extraíndo diferentes amostras uniformes a cada iteração  $k$  no Algoritmo 1. Sem essa amostragem, a utilização de memória e o tempo de execução esperado para um único lote se tornam imprevisíveis, podendo atingir  $O(|V|)$  no pior cenário. Em contraste, a complexidade espacial e temporal por lote para o GraphSAGE é fixa em  $O(\prod_{i=1}^K S_i)$ , onde  $S_i \in \{1, \dots, K\}$  e  $K$  são constantes definidas pelo usuário.

Embora o Algoritmo 1 pressuponha que todo o grafo esteja disponível em memória, esse cenário pode se tornar inviável em aplicações com grafos de grande escala. Para contornar essa limitação, o GraphSAGE adota uma abordagem baseada em *minibatches*, permitindo que o modelo seja treinado de forma eficiente mesmo em grandes volumes de dados.

Para possibilitar o uso de técnicas de otimização baseadas em gradiente descendente estocástico (*stochastic gradient descent*), o algoritmo é adaptado para suportar a propagação direta (*forward*) e reversa (*backward*) em *minibatches* de nós e arestas. O foco aqui está na propagação direta, que é análoga ao Algoritmo 1. Nessa abordagem, o *minibatch*  $B$  contém os nós para os quais se deseja gerar representações vetoriais.

Inicialmente, realiza-se a amostragem antecipada das vizinhanças necessárias até uma profundidade  $K$ . Em seguida, executa-se o loop interno de propagação (correspondente à linha 3 do Algoritmo 1), mas, em vez de percorrer todos os nós do grafo, o algoritmo calcula apenas as representações estritamente necessárias para satisfazer a recursão em cada nível. Essa estratégia reduz significativamente o custo computacional e torna o modelo compatível com métodos tradicionais de treinamento supervisionado. O Algoritmo 2, a seguir, apresenta o pseudocódigo correspondente à abordagem baseada em *minibatches*.

A ideia central por trás da propagação em *minibatch* no GraphSAGE é realizar, de forma antecipada, a amostragem de todos os nós necessários para o cálculo das representações. No Algoritmo 2, as linhas 2 a 7 correspondem à etapa de amostragem. Cada conjunto  $B^k$  armazena os nós necessários para computar as representações dos nós em  $B^{k+1}$ , ou seja, aqueles que serão processados na iteração  $(k + 1)$  do Algoritmo 1. Já as linhas 9 a 15 representam a etapa de agregação, que segue de forma quase idêntica à propagação em *batch*.

---

**Algoritmo 2:** Algoritmo de propagação direta em minibatch do GraphSAGE
 

---

Entrada: Grafo  $G(V, E)$ ; características de entrada  $\{x_v, \forall v \in V\}$ ; profundidade  $K$ ; matrizes de pesos  $W_k, \forall k \in \{1, \dots, K\}$ ; não-linearidade  $\sigma$ ; funções de agregação diferenciáveis  $AGGREGATE_k, \forall k \in \{1, \dots, K\}$ ; funções de amostragem de vizinhança  $N_k: v \rightarrow 2^V, \forall k \in \{1, \dots, K\}$

Saída: Representações vetoriais  $z_v$  para todos  $v \in V$

```

1  $B^k \leftarrow B$ ;
2 for  $k = K \dots 1$  do
3    $B^{k-1} \leftarrow B^k$ ;
4   for  $u \in B^k$  do
5      $B^{k-1} \leftarrow B^{k-1} \cup N_k(u)$ 
6   end
7 end
8  $h_u^0 \leftarrow x_v, \forall v \in B^0$ ;
9 for  $k = 1 \dots K$  do
10  for  $u \in B^k$  do
11     $h_{N(u)}^k \leftarrow AGGREGATE_k(\{h_u^{k-1}, \forall u' \in N(u)\})$ ;
12     $h_u^k \leftarrow \sigma(W^k \cdot CONCAT(h_u^{k-1}, h_{N(u)}^k))$ 
13     $h_v^k \leftarrow h_u^k / \|h_u^k\|_2$ ;
14  end
15 end
16  $z_u \leftarrow h_u^K, \forall u \in B$ 

```

---

Durante esse processo, a representação de um nó  $u \in B^k$  pode ser calculada na iteração  $k$ , pois tanto sua própria representação na iteração  $k - 1$  quanto as representações de seus vizinhos amostrados nesse mesmo nível já foram computadas no laço anterior. Dessa forma, o algoritmo evita o cálculo desnecessário de embeddings para nós que não pertencem ao *minibatch* atual e que não serão utilizados durante a iteração corrente do treinamento.

A amostragem da vizinhança é realizada por meio de uma função determinística  $N_k(u)$ , que define uma amostra aleatória dos vizinhos de um nó  $u$ . Assume-se que a aleatoriedade já foi previamente computada, tornando o processo reprodutível e eficiente. O índice  $k$  na função indica que as amostras são independentes entre diferentes níveis de profundidade. Neste trabalho, adota-se uma função de amostragem uniforme, com reposição nos casos em que o número de vizinhos desejado é superior ao grau do nó.

### 3.2.1.2. Aprendizado dos parâmetros do GraphSAGE

Para aprender representações úteis e preditivas em um ambiente totalmente não supervisionado, é aplicada uma função de perda baseada em grafo às representações de saída,  $z_u; \forall u \in V$ . As matrizes de peso,  $W_k; \forall k \in \{1, \dots, K\}$ , e os parâmetros das funções agregadoras são ajustados por meio de descida de gradiente estocástica. A função de perda baseada em grafo estimula a similaridade entre as representações de nós próximos, enquanto assegura que as representações de nós distintos sejam altamente diferenciadas. A Equação 7 ilustra esse processo.

$$J_G(Z_u) = -\log(\sigma(Z_u^T Z_v)) - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log(\sigma(-Z_u^T Z_{v_n})), \quad (7)$$

onde  $v$  representa um nó que co-ocorre próximo a  $u$  em uma caminhada aleatória de comprimento fixo,  $\sigma$  é a função sigmoide,  $P_n$  refere-se a uma distribuição de amostragem negativa e  $Q$  indica o número de amostras negativas. É importante ressaltar que, diferentemente de abordagens anteriores de *embeddings*, as representações  $z_u$  utilizadas nesta função de perda são geradas a partir das características da vizinhança local de um nó, em vez de treinar um embedding exclusivo para cada nó (por meio de uma busca de *embedding*).

Esse ambiente não supervisionado simula cenários nos quais as características dos nós são disponibilizadas para aplicações de aprendizado de máquina subsequentes, seja como um serviço ou em um repositório estático. Quando as representações são destinadas a uma tarefa específica, a perda não supervisionada (Equação 1) pode ser substituída ou aprimorada por um objetivo específico da tarefa (como a perda de entropia cruzada).

### 3.2.1.3. Arquitetura de agregadores

Diferente do aprendizado de máquina em grades N-D (como sentenças, imagens ou volumes 3D), os vizinhos de um nó em um grafo não seguem uma ordem natural. Por isso, as funções de agregação no Algoritmo 1 precisam lidar com um conjunto de vetores sem ordem definida. O ideal é que essas funções sejam simétricas (ou seja, que a ordem das entradas não importe), mas ainda assim sejam treináveis e com boa capacidade de representar informações. A simetria garante que o modelo de rede neural funcione bem com conjuntos de características dos nós, independentemente da ordem dos vizinhos. A seguir, são apresentadas três funções agregadoras, que podem ser aplicadas nesse contexto.

**Mean aggregator:** A primeira função agregadora é o operador de média, no qual se calcula a média elemento a elemento dos vetores em  $\{h_u^{k-1} ; \forall u \in N(v)\}$ , conforme Equação 8. O *mean aggregator* pode ser visto como uma aproximação da operação convolucional da estrutura transdutiva da Rede Neural Convolucional em Grafos (*Graph Convolutional Network* - GCN) (Kipf and Welling, 2016). Em particular, é possível derivar uma variante indutiva da abordagem GCN ao substituir as linhas 4 e 5 do Algoritmo 1 pela seguinte forma.

$$h_v^k \leftarrow \sigma(W \cdot \text{MEAN}(\{h_v^{k-1}\} \cup \{h_u^{k-1}, \forall u \in N(v)\})) \quad (8)$$

Esse agregador modificado baseado em média é denominado convolucional, pois representa uma aproximação linear rudimentar de uma convolução espectral localizada. Uma diferença fundamental entre este agregador convolucional e os outros agregadores propostos é que ele não realiza a operação de concatenação na linha 5 do Algoritmo 1. Em outras palavras, o agregador convolucional não combina a representação da camada anterior do nó,  $h_v^{k-1}$ , com o vetor agregado da vizinhança,  $h_{N(v)}^k$ . Essa concatenação pode ser interpretada como uma forma simples de "conexão de salto" entre as distintas "profundidades de busca" ou "camadas" do algoritmo GraphSAGE, o que pode resultar em melhorias no desempenho.

**LSTM aggregator:** Trata-se de um agregador mais complexo, fundamentado em uma arquitetura de memória de longo e curto prazo (*long short-term memory* – LSTM) (Hochreiter & Schmidhuber, 1997). Em comparação ao agregador de média, as LSTMs apresentam uma capacidade expressiva superior. Isso se deve ao fato de que as LSTMs são capazes de capturar dependências temporais ou relacionais ao longo de uma sequência, utilizando mecanismos internos de memória (como a célula de estado e as portas de entrada, esquecimento e saída) que permitem reter ou descartar informações conforme a relevância ao longo do tempo ou da ordem da sequência. Contudo, é importante destacar que as LSTMs não possuem simetria intrínseca, ou seja, não são invariantes a permutações, uma vez que processam suas entradas de forma sequencial. Para adaptar as LSTMs a conjuntos desordenados, aplica-se a arquitetura a uma permutação aleatória dos vizinhos do nó.

**Pooling aggregator:** é um agregador simétrico e treinável. Nessa abordagem de *pooling*, cada vetor de vizinho é processado individualmente por uma rede neural totalmente conectada. Após essa transformação, aplica-se uma operação de *max-pooling* elemento a elemento para agregar as informações do conjunto de vizinhos. Esse procedimento pode ser visto na Equação 9.

$$AGREGATE_K^{POOL} = \max(\{\sigma(W_{pool}h_{u1}^k + b), \forall u_i \in N(v)\}) \quad (9)$$

onde  $\max$  indica o operador de máximo elemento a elemento, e  $\sigma$  representa uma função de ativação não linear. O perceptron multicamadas pode ser interpretado como um conjunto de funções que extraem características das representações dos nós vizinhos. Ao aplicar o operador de *max pooling* a essas características, o modelo captura diferentes aspectos do conjunto de vizinhos. É importante destacar que qualquer função vetorial simétrica poderia, em princípio, substituir o operador de *max pooling* (como, por exemplo, uma média elemento a elemento).

### 3.3. Representação de grafos: matrizes de adjacência e matrizes ponderadas

A representação de grafos por meio de matrizes é um passo fundamental na implementação de GNNs, pois influencia diretamente a forma como as informações estruturais são propagadas e processadas. Dentre as formas mais comuns de representação matricial estão a matriz de adjacência binária e a matriz ponderada. A escolha entre essas estruturas afeta tanto a fidelidade com que a topologia da rede é representada quanto o desempenho dos modelos de aprendizado baseados em grafos.

#### 3.3.1. Matriz de adjacência (binária)

A matriz de adjacência binária é uma forma clássica e amplamente utilizada para representar a estrutura de um grafo. Em um grafo não direcionado com  $N$  nós, essa matriz  $A \in R^{N \times N}$  é definida pela Equação 10.

$$A_{ij} = \begin{cases} A_{ij}, & \text{se houver uma aresta entre os nós } i \text{ e } j \\ 0, & \text{caso contrário} \end{cases} \quad (10)$$

Essa representação binária é simples e eficiente, sendo particularmente adequada para grafos esparsos e não ponderados, como redes sociais e alguns tipos de redes computacionais (Wu et al., 2021). No entanto, para aplicações em que a intensidade ou natureza da relação entre os nós é relevante, como em redes físicas de infraestrutura, essa simplificação pode resultar na perda de informações importantes.

### 3.3.2. Matrizes ponderadas

As matrizes ponderadas, por sua vez, incorporam informações adicionais às conexões entre os nós, atribuindo pesos às arestas com base em atributos relevantes, como distância física, condutância, capacidade de fluxo ou outros critérios específicos do domínio. A matriz ponderada  $W_{ij} \in R^{N \times N}$  é definida pela Equação 11.

$$W_{ij} = \begin{cases} w_{ij}, & \text{se houver uma aresta entre os nós } i \text{ e } j \\ 0, & \text{caso contrário} \end{cases} \quad (11)$$

A principal diferença entre as matrizes binárias e as ponderadas está, portanto, na expressividade. Enquanto a matriz binária codifica apenas a existência de conexões, a matriz ponderada quantifica a intensidade ou relevância dessas conexões (Kipf & Welling, 2017; Zhou et al., 2020).

## 4. MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais e métodos empregados nesta pesquisa, para o estudo da aplicação de algoritmos de GNN para a detecção e localização de vazamentos em RDAs. Primeiramente, será discutida a modelagem e simulação das arquiteturas das RDAs utilizadas, bem como a aquisição dos seus dados. Em seguida, será descrito o processo de aplicação da GNN utilizada, apresentando os hiperparâmetros adotados, além das métricas de avaliação empregadas para mensurar a performance do algoritmo. O conteúdo aqui abordado tem como referência os trabalhos desenvolvidos em Rolle (2024), Rolle et al. (2022), sendo todo o procedimento do modelamento das RDAs baseado em Tomazini et al. (2024).

### 4.1. Simulação de cenários via EPANET

A utilização de dados simulados justifica-se por razões práticas, sobretudo pelo alto custo da instrumentação de redes com dispositivos de medição remotos e pela inviabilidade operacional de induzir vazamentos de diferentes tamanhos e locais em uma rede de distribuição real. Além disso, a aplicação de algoritmos de ML na detecção de vazamentos exige grandes volumes de dados para o treinamento. Dessa forma, a simulação computacional de cenários de RDAs representa uma alternativa mais viável e econômica, permitindo a criação de condições diversas e controladas para a geração dos dados, possibilitando o treinamento dos algoritmos de ML.

#### 4.1.1. Configuração da rede de distribuição de água

Para que o modelo simule uma RDA real, é fundamental reunir informações detalhadas sobre a tubulação existente, incluindo o comprimento, diâmetro, material dos canos, e a presença de curvas, válvulas, e outros elementos de controle. Em um modelo, cada ponto de consumo e/ou monitoramento é representado por um nó. Além disso, é essencial contar com estimativas do consumo de água na rede e com dados dos níveis de pressão provenientes do sistema de abastecimento.

A construção de uma RDA no ambiente computacional do EPANET envolve os seguintes passos:

- Estabelecer as unidades de medida;
- Adicionar os reservatórios de nível fixo (RNF);

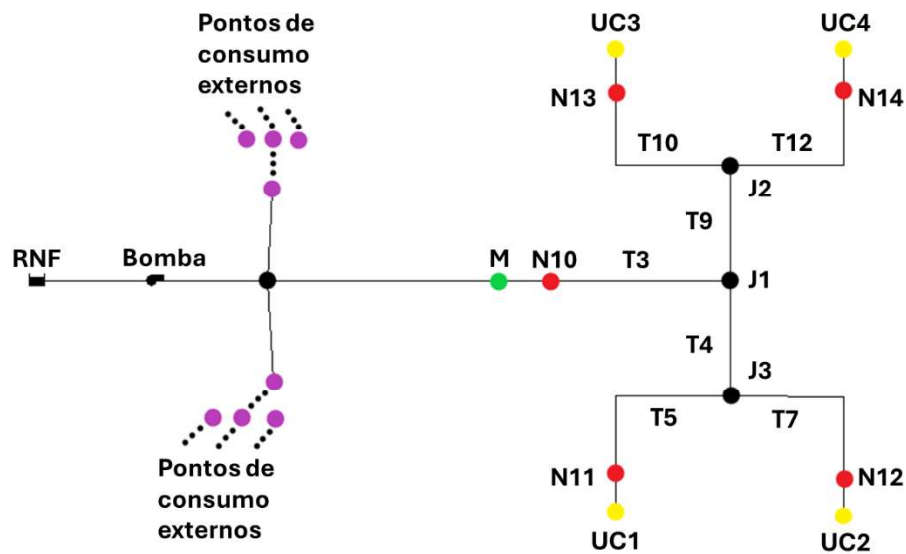
- Adicionar os nós e reservatórios de nível variável (RNV);
- Adicionar os trechos (canos e tubulação que interliga os nós);
- Adicionar as bombas e respectivas curvas de rendimento;
- Criar a curva da bomba (condições de funcionamento);
- Definir as opções de temporização: tempo total da simulação, intervalo de cálculo hidráulico e intervalo para geração de relatório;
- Criar padrões de consumo e atribuí-los aos nós;
- Editar os nós, atribuindo a eles padrões de consumo e valores de consumo-base.

No EPANET, cada nó funciona como uma entidade genérica, permitindo diferentes interpretações. Nas simulações de RDAs abordadas neste estudo, os nós são divididos nas seguintes categorias:

- (i) **Nós de junção:** representam as ramificações da rede, como conexões em "T", onde não ocorre consumo ou medição de variáveis;
- (ii) **Nós de medição:** são geralmente posicionados na entrada da rede e responsáveis por registrar variáveis;
- (iii) **Nós de consumo:** correspondem aos pontos de utilização normal de água e incluem a medição individual de consumo; e
- (iv) **Nós de vazamento:** são introduzidos especificamente para simular eventos de vazamento.

Na Figura 5, que representa uma RDA de um condomínio simples com quatro residências, é possível visualizar todas essas características. Nessa configuração, o nó de medição, que está em verde, indica o medidor de entrada principal, enquanto os nós de consumo, em amarelo, representam as quatro casas, cada qual equipada com um medidor individual que registra seu consumo específico. Os nós de junções estão em preto e os nós, que representam os vazamentos, estão em vermelho.

Figura 5 - Ilustração de uma RDA hipotética representando um condomínio fictício com quatro casas



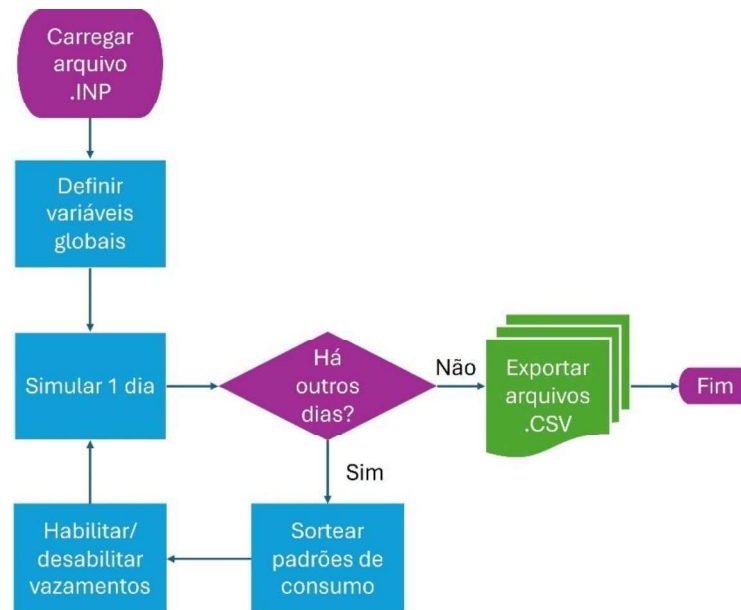
Fonte: Adaptado de Tomazini et al. (2024)

O sistema de abastecimento de água é simulado por meio de um RNF associado a uma bomba, representando a Estação de Tratamento de Água (ETA) e os sistemas de bombeamento utilizados pelas empresas de saneamento. Pontos de consumo adicionais, em roxo, podem ser distribuídos pela rede para reproduzir a demanda de água no restante do sistema, não monitorado, capturando assim as variações de consumo típicas nas RDAs. O modelo também incorpora a natureza estocástica das demandas dos usuários, representando-a por padrões variados de consumo que são atribuídos aleatoriamente a cada dia simulado, correspondente a um período de 24 horas.

#### 4.1.2. Aquisição dos dados

Após a criação e parametrização da rede no EPANET, o modelo pode ser exportado no formato ".inp", compatível com o EPANET-MATLAB *toolkit*. Isso permite a elaboração de *scripts* que automatizam as simulações. O processo realizado no MATLAB é sintetizado no fluxograma da Figura 6, e inicia-se com o carregamento do arquivo ".inp". Em seguida, são definidas as variáveis de interesse (como nós de medição), padrões adicionais de consumo para sorteio, além da configuração do programa de simulação, definindo número de dias a serem simulados e a ocorrência de vazamentos em determinados dias. Ao final, os conjuntos de dados gerados são exportados no formato CSV (*Comma-Separated Values*) para armazenamento e processamento posterior.

Figura 6 - Fluxograma da rotina de simulação implementada utilizando o EPANET-MATLAB *toolkit*.



Fonte: Adaptado de Rolle et al. (2022)

#### 4.1.3. Geração dos cenários

Para a geração de dados, em Rolle (2024) foram desenvolvidos dois cenários de RDAs hipotéticas no aplicativo EPANET, que aqui também foram reproduzidos para análise da aplicação do GraphSAGE. Embora cada cenário tenha suas especificidades, como número de nós, configuração da rede e quantidade de pontos de consumo, alguns parâmetros foram padronizados para todos os casos, conforme detalhado na Tabela 1. Os pontos de consumo externos à RDA no setor monitorado têm o objetivo de representar o consumo no restante do sistema de abastecimento urbano, induzindo variações de pressão ao longo do dia devido aos diferentes perfis de demanda.

Tabela 1 – Parâmetro comuns dos modelos de RDAs gerados

| Parâmetro                  | Valor    |
|----------------------------|----------|
| Diâmetro da tubulação      | 2"       |
| Material da tubulação      | PVC      |
| Pontos de consumo externos | 30       |
| Número de RNFs             | 1        |
| Número de RNVs             | 0        |
| Intervalo de medição       | 10 min   |
| Tempo de simulação         | 24 horas |

Fonte: Adaptado de Rolle (2024).

No EPANET, cada nó e segmento de tubulação recebe um identificador numérico (nós N1, N2...; trechos T1, T2...), o qual também é utilizado no *toolkit* do MATLAB para a coleta de dados exclusivamente nos pontos especificados como nós de medição. A saída gerada pelo EPANET é chamada de “relatório” da simulação, uma tabela com dados de todos os nós e segmentos da rede. Para facilitar o gerenciamento dessas informações, foram desenvolvidos *scripts* no MATLAB que organizam as tabelas de saída de forma padronizada, contendo apenas os dados dos pontos de medição. Esses dados incluem:

- Estampa de tempo (*timestamp*);
- Pressão em cada ponto de medição;
- Coordenadas de cada ponto de medição;
- Variáveis que indicam o local de vazamento ativo (quando houver).

#### **4.1.4. Conversão da estrutura da RDA em grafo**

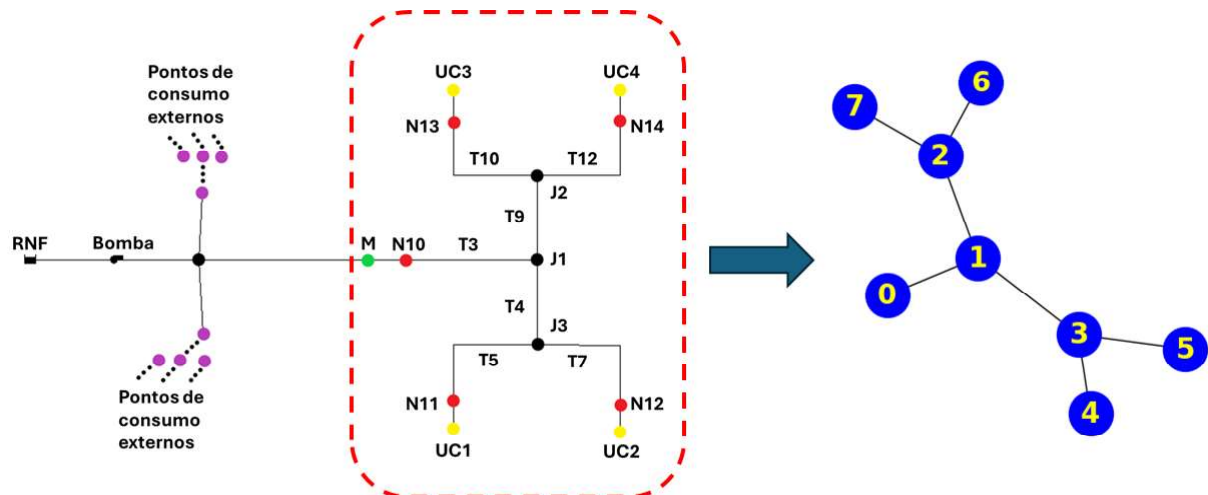
Os dados de entrada das GNNs incluem duas componentes principais: a matriz de representação da estrutura da rede e a matriz de atributos nodais. A primeira pode assumir a forma de uma matriz de adjacência (binária) ou de uma matriz de distâncias (com valores métricos entre os nós), e descreve a estrutura do grafo, indicando as conexões entre os nós, o que representa a topologia da rede e permite que o modelo aprenda as relações entre os elementos conectados. A matriz de atributos, por sua vez, contém informações específicas de cada nó, como medições de pressão, vazão ou outras características relevantes para o problema modelado. Juntas, essas matrizes fornecem uma visão abrangente da rede e permitem que a GNN capture tanto as características locais quanto as dependências estruturais do sistema.

Para obter essas duas componentes, é necessário converter a estrutura da RDA simulada no *software* EPANET em um grafo, onde as instalações de tratamento, armazenamento, bombeamento e pontos de consumo poderão ser representados por nós, com atributos específicos, enquanto os canos e tubulações de distribuição poderão ser representados como arestas.

Vale resaltar que as simulações das RDAs modeladas incluem pontos específicos para representar os vazamentos, que podem ser ativados ou desativados sistematicamente. Esses pontos de vazamento não são representados no grafo, pois os vazamentos podem ocorrer em qualquer posição na rede e não fazem parte do comportamento normal do sistema. A Figura 7 ilustra o modelo esquemático de uma das RDAs modeladas e seu grafo correspondente. Para o

caso em questão, a conversão da RDA em grafo ocorre apenas na parte do sistema em que é realizado o monitoramento, correspondendo à área circulada em vermelho.

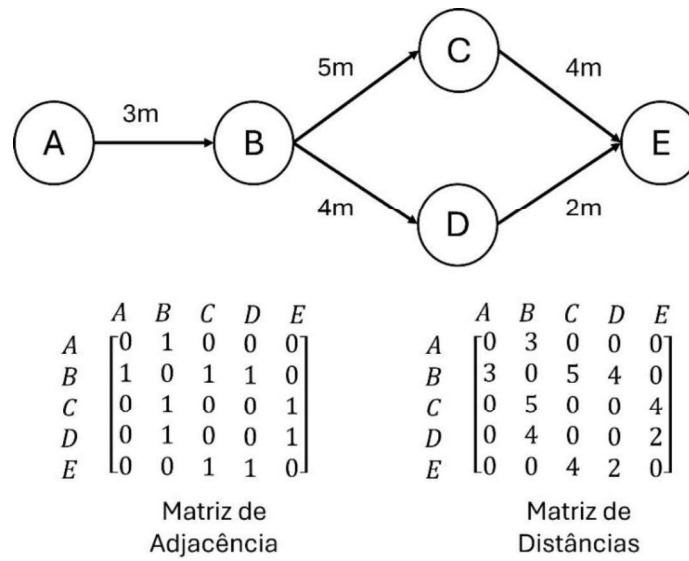
Figura 7 - Conversão da estrutura da RDA em grafo



Fonte: Adaptado de Rolle et al. (2022)

A estrutura do grafo pode ser representada matematicamente por meio da matriz de adjacência, que descreve as conexões entre os pontos de medição e consumo da rede. Os valores dessa matriz indicam se existe uma ligação direta entre os nós, onde “0” significa ausência de conexão e “1” indica presença de conexão. Alternativamente, a estrutura também pode ser representada pela matriz de distâncias, que reflete o comprimento dos trechos de tubulação que conectam os nós da rede, podendo ser utilizada como a ponderação da relação das arestas, de forma direta ou inversamente proporcional ao seu valor. A Figura 8 ilustra essa estrutura em detalhe.

Figura 8 - Exemplo de grafo com suas matrizes de adjacência e distâncias



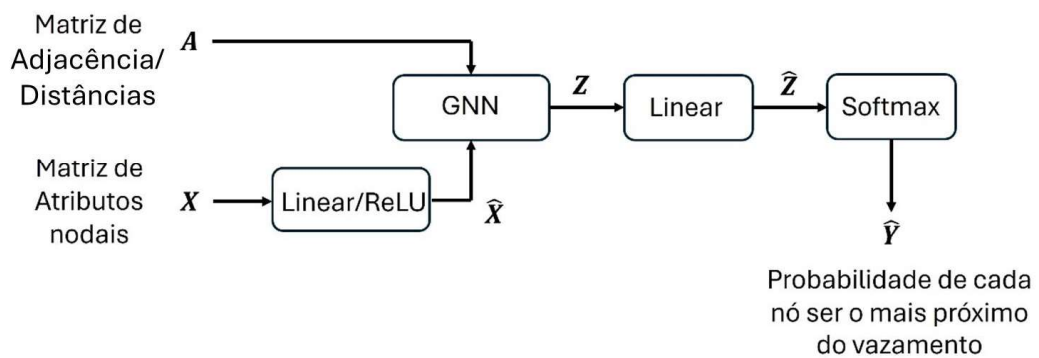
Fonte: Autoria própria

Neste estudo, foram utilizadas tanto a matriz de adjacência quanto a matriz de distâncias para representar as conexões entre os nós do grafo, com o objetivo de comparar o impacto de cada abordagem no desempenho do modelo de GNN.

#### 4.1.5. Fluxograma da aplicação do modelo de GNN

O fluxograma da aplicação do modelo de GNN aqui utilizado é apresentada na Figura 9, sendo o mesmo aplicado no trabalho desenvolvido por Rolle et al. (2022).

Figura 9 - Fluxograma de aplicação do modelo de GNN aplicado para detecção e localização de vazamentos.



Fonte: Adaptado de Rolle et al. (2022)

As entradas do modelo são a matriz de adjacência ou de distâncias e a matriz contendo os atributos dos nós, conforme apresentado na Equação 12. Neste estudo, esses atributos correspondem às medições de pressão em cada nó. A primeira camada aplica uma combinação linear padrão, seguida de uma função de ativação ReLU (*Rectified Linear Unit*).

$$\hat{X} = ReLU(W_{in}X + b_{in}) \quad (12)$$

onde,  $W_{in} \in \mathbb{R}^{m \times r}$  é a matriz de pesos de entrada,  $b_{in}$  são os *biases* de entrada, e  $ReLU(\cdot)$  é uma função aplicada elemento a elemento, tal que  $ReLU(x) = \max(0, x)$ . A variável  $r$  indica o número de atributos de entrada, enquanto  $m$  define o tamanho dos estados ocultos. Um maior número de estados ocultos pode ampliar a capacidade de aprendizado do modelo, mas também aumenta o custo computacional e o risco de *overfitting* (sobreajuste aos dados de treinamento), podendo prejudicar a capacidade de generalização do modelo.

Em seguida, uma nova representação  $Z$  do grafo é gerado pelo modelo de GNN, a partir de  $\hat{X}$  e  $A$ , segundo Equação 13.

$$Z = GNN(\hat{X}, A) \quad (13)$$

O vetor de estados final  $Z_v$ , de cada nó, armazena informações de um certo número de nós vizinhos. O alcance dessa vizinhança é um dos hiperparâmetros do modelo, denominado "passos de propagação" (ou *propagation steps*), que pode ser ajustado para otimizar o desempenho do modelo.

Na sequência, uma pontuação  $\hat{Z}_v$  é atribuída a cada vetor de estado de  $Z_v$  utilizando uma combinação linear, tal como apresentado na Equação 14.

$$\hat{Z} = W_{out}Z + b_{out} \quad (14)$$

onde  $W_{out} \in \mathbb{R}^m$  são os pesos da saída e  $b_{out}$  são os *biases* da saída.

Por fim, de acordo com a Equação 15, a função *softmax* faz a normalização das saídas, distribuindo suas respectivas probabilidades, onde ao nó de maior índice, é atribuído a localidade do vazamento, por entender que ele é quem se encontra mais próximo a ele.

$$\hat{y}_v = \frac{e^{\hat{Z}_v}}{\sum_{u \in \gamma} e^{\hat{Z}_u}} \quad (15)$$

#### 4.1.6. Processamento de dados

A etapa de processamento envolve preparação dos dados para a execução do treinamento e teste dos algoritmos de ML aqui aplicados. Neste estudo, tanto a etapa de processamento de dados, quanto os algoritmos, foram implementados em Python utilizando o ambiente *Visual Studio Code* (VS Code), sendo os algoritmos baseados na biblioteca PyTorch. O procedimento completo para esta execução compreende várias etapas, que serão detalhadas a seguir.

A etapa inicial envolve o carregamento dos dados obtidos das simulações dos modelos de RDAs, que consiste em adquirir os dados armazenados (arquivos CSV das simulações) e formatá-los de acordo com os requisitos da biblioteca PyTorch. Para isso, são realizadas as seguintes ações:

1. Carregar os arquivos de dados;
2. Converter as tabelas para o formato de dataframe, para facilitar a manipulação dos dados (como nomeação de colunas, normalização, indexação e operações em matrizes);
3. Realizar a preparação e normalização dos dados;
4. Identificar às colunas (*features*) dos diferentes tipos de dados;
5. Identificar os cenários de vazamento e atribuir rótulos (*labels*) apropriados (por exemplo, sem vazamento, vazamento no ponto A, vazamento no ponto B, etc.) para uso no treinamento;
6. Gerar tabelas distintas para os identificadores (*features*) e os rótulos (*labels*);
7. Dividir os dados em conjuntos de treinamento e teste;
8. Converter os conjuntos para o formato tensor, compatível com a biblioteca PyTorch;
- 9.1. Criar a matriz de adjacência binária que representa a rede.
- 9.2. Construir a matriz de ponderação baseada nas distâncias da rede e realizar a normalização de seus dados.

Na etapa de preparação e normalização dos dados, a pressão medida em cada ponto da rede foi ajustada subtraindo-se o valor de pressão registrado no medidor de entrada da rede (representado em verde no mapa de cada rede). Esse procedimento visa normalizar os dados em relação a um ponto de referência comum, eliminando o viés absoluto e destacando as diferenças relativas de pressão na rede.

#### 4.1.7. Parâmetros utilizados para os algoritmos

Após o carregamento e preparação dos dados, é possível executar o algoritmo principal do GNN, cujo modelo é construído com base nas matrizes de adjacência/distância e de atributos dos nós. Além disso, são definidos os seguintes parâmetros para o algoritmo:

- **Número de vizinhos (*Number of neighbors*):** quantidade máxima de nós vizinhos a serem amostrados e utilizados durante o processo de agregação para atualizar a representação (embedding) de um nó-alvo. Essa amostragem é uma característica específica do algoritmo GraphSAGE, permitindo o uso eficiente de vizinhanças em grafos grandes por meio de uma abordagem estocástica.
- **Passes de propagação (*Propagation steps*):** determina o alcance máximo de propagação de informações entre os nós (definindo o tamanho da vizinhança). Um valor baixo pode limitar o aprendizado das dependências entre nós distantes, enquanto valores mais altos aumentam o custo computacional e a demanda de memória durante o treinamento;
- **Tamanho das camadas escondidas (*Hidden layer*):** indica o número de neurônios nas camadas ocultas, influenciando tanto a capacidade de aprendizado do modelo quanto o custo computacional;
- **Tamanho do lote de treinamento (*Batch size*):** número de amostras de treinamento processadas em cada iteração;
- **Métodologia de divisão dos dados para validação:** consiste em escolher qual método será utilizado para dividir o conjunto de dados em treinamento e teste, para validar o modelo;
- **Função de custo (*Loss function*):** utilizada para medir o desempenho do modelo durante o treinamento;
- **Função de otimização (*Optimizer*):** responsável por ajustar os parâmetros do modelo para melhorar o aprendizado;
- **Número de épocas (*Number of epochs*):** quantidade de vezes que os dados de treinamento são completamente fornecidos para o algoritmo aprender;

#### 4.2. Métodos de avaliação da performance do modelo

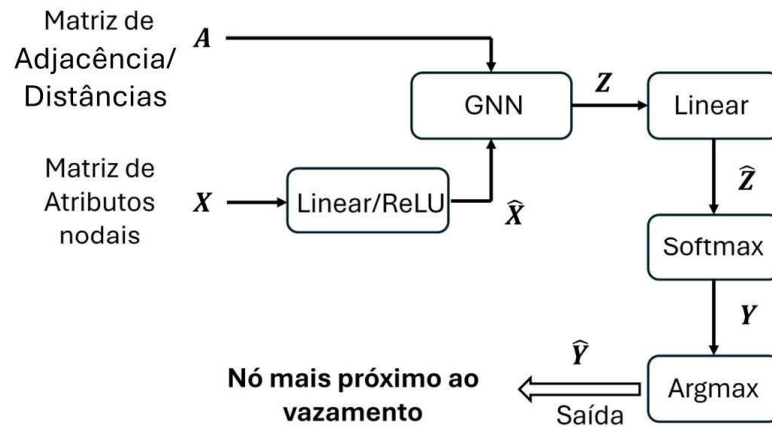
Durante o treinamento da rede, as saídas do modelo são comparadas com os resultados esperados (baseados nos identificadores definidos previamente), utilizando uma função de custo (*loss function*). Essa função de custo quantifica o quão bem o modelo está se saindo em

relação aos dados de treinamento, sendo o objetivo do treinamento minimizar os resultados dessa função, ajustando os pesos da rede através de algoritmos de otimização.

Existem várias funções de perda que podem ser utilizadas, dependendo da tarefa. Para problemas de classificação, como no caso de redes neurais de grafo, a função de custo mais comum é a entropia cruzada, que mede a diferença entre a distribuição de probabilidade prevista pelo modelo e a distribuição real das classes.

Para calcular a acurácia do modelo, a distribuição de probabilidades gerada pela função Softmax será processada pela função Argmax, atribuindo valor 1 ao nó com a maior probabilidade e 0 aos demais. Essa incorporação é ilustrada na Figura 10.

Figura 10 - Fluxograma final da aplicação do modelo de GNN aplicado para detecção e localização de vazamentos.



Fonte: Adaptado de Rolle et al. (2022).

Ao final do treinamento, são gerados gráficos que mostram a evolução da função de custo e da acurácia durante o treinamento e teste, permitindo uma avaliação do desempenho do modelo de GNN.

A acurácia é uma métrica que indica a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões, sendo amplamente utilizada como métrica para avaliar modelos de classificação, como é o caso da tarefa em questão, sendo calculada conforme mostrado na Equação 16.

$$Acurácia = \frac{\text{Número de classificações corretas}}{\text{Número total de amostras}} \quad (16)$$

No entanto, sua aplicação apresenta limitações significativas, especialmente em contextos onde os dados ou os objetivos do modelo possuem características específicas que requerem uma análise mais detalhada.

Por ser uma métrica agregada, a acurácia não oferece detalhes sobre o desempenho do modelo em cada classe individualmente, não fazendo a distinção entre erros de diferentes tipos, como falsos positivos e falsos negativos, nem revela se ele está cometendo mais erros em classes específicas. Isso dificulta a identificação de padrões de falhas que poderiam ser corrigidos para melhorar sua robustez.

Por essa razão, confiar exclusivamente na acurácia pode levar a avaliações equivocadas. Em contextos como esses, é essencial complementá-la com a utilização de outras métricas que possam oferecer uma visão mais detalhada do problema, como a métrica F1-Score, dado pela Equação 19, que realiza a média harmônica entre a Precisão, que é a proporção de previsões corretas entre as feitas para uma classe, dado pela Equação 17, e o *Recall*, que é a proporção de itens de uma classe corretamente identificados, dado pela Equação 18. Isso garantirá uma avaliação mais robusta e alinhada aos objetivos reais da aplicação do modelo.

$$Precisão = \frac{TP}{TP + FP} \quad (17)$$

onde *TP* são os Verdadeiros Positivos e *FP* são os Falsos Positivos.

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

onde *FN* são os Falsos Negativos.

$$F1_{score} = 2 \cdot \frac{Recall \times Precisão}{Recall + Precisão} \quad (19)$$

## 5. RESULTADOS E DISCUSSÃO

Neste capítulo serão apresentados os dois modelos de RDAs hipotéticas utilizadas para o estudo da aplicação do modelo GraphSAGE para detecção e localização de vazamentos. Como já mencionado, elas foram utilizadas em trabalhos anteriores, com o intuito de validar a utilização de GNNs para a tarefa. Elas representam cenários de aplicação de pequeno porte, no caso, instalações de condomínios residenciais supridos pelos sistemas públicos de saneamento básico. Os resultados obtidos com os dois modelos serão comparados àqueles do modelo de GNN proposto por Rolle (2024), visando avaliar o potencial desta nova proposta.

### 5.1. Descrição dos experimentos

Para a aplicação do modelo GraphSAGE, com três agregadores distintos, na tarefa de detecção e localização de vazamentos em RDAs, foram empregadas as mesmas arquiteturas de rede utilizadas por Rolle (2024), sendo elas do tipo ramificada, simuladas através do *software* EPANET e manipuladas para a geração dos *datasets* utilizando o EPANET-MATLAB *toolkit*, conforme descritos no capítulo 4. Cada uma dessas arquiteturas possui um grafo equivalente, composto por nós, que representam pontos de medição e junções, e arestas que correspondem aos canos e tubulações que os conectam. Além disso, nessas arquiteturas, pontos adicionais foram configurados para representar os possíveis vazamentos, ativados em situações específicas, conforme o programa de simulação desenvolvido. Durante os experimentos, os dados de pressão são coletados nos nós de medição de cada rede, bem como nas junções, incluindo bifurcações, junções em “T” e outras configurações.

Para validar os resultados obtidos, a performance dos modelos empregados neste estudo será comparada à do modelo de GNN utilizado por Rolle (2024), que servirá como *benchmark* para esta pesquisa. Essa comparação fornecerá uma base sólida para a avaliação da eficácia do modelo GraphSAGE na execução desta tarefa.

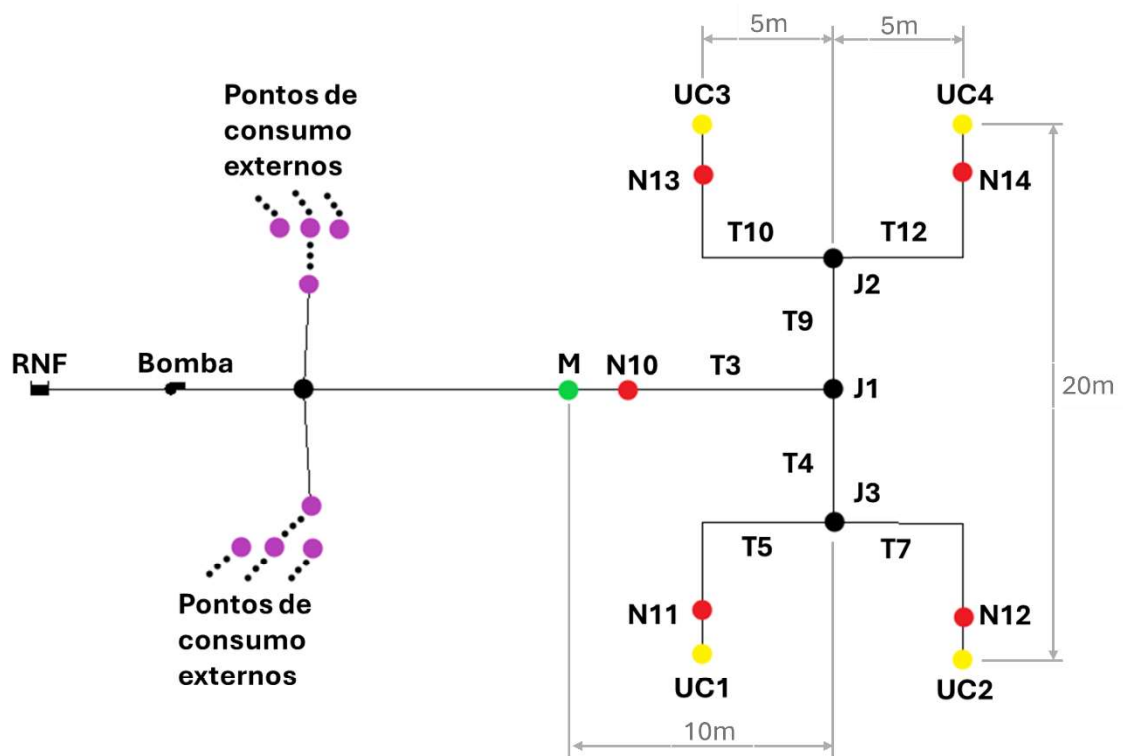
### 5.2. Cenário 1

A Figura 11 apresenta a primeira rede, que representa um conjunto residencial com quatro casas (indicadas pelos pontos amarelos), cada uma equipada com um medidor de entrada. A rede também conta com um medidor posicionado na entrada principal da tubulação (ponto verde). Para simular o impacto do consumo urbano, foram adicionados 30 pontos de consumo externos (pontos roxos), representando demandas adicionais que influenciam o abastecimento. Os pontos pretos marcam as junções (como ligações em T), enquanto os pontos vermelhos são

reservados para simulação de vazamentos. As distâncias mais importantes entre os componentes são destacadas por cotas (em proporções não exatas). Este modelo incorpora 16 padrões diários de consumo, atribuídos aleatoriamente às unidades consumidoras nas simulações diárias. Outros 11 padrões representam o consumo dos pontos externos à rede monitorada, também aplicados de maneira randômica para simular demandas urbanas variadas.

Os vazamentos são representados na rede por meio da inserção do coeficiente de emissão nos nós correspondentes, conforme descrito pela Equação 6. Dessa forma, a vazão dos vazamentos varia dinamicamente durante a simulação, acompanhando as variações de pressão do sistema. Nesta simulação, os vazamentos resultaram em uma vazão média de aproximadamente 3 LPM, podendo ocorrer continuamente ao longo das 24 horas do dia ou apenas em determinados períodos.

Figura 11 - RDA do primeiro cenário.



Fonte: Adaptado de Rolle et al. (2022)

A sequência de simulações é dada por:

- 100 simulações sem vazamento;
- 50 simulações com vazamentos de dia inteiro no nó N10;

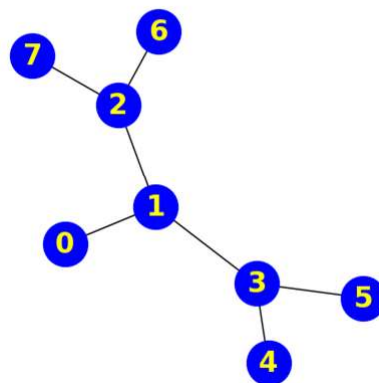
- 50 simulações com vazamentos de meio no nó N10;
- 50 simulações com vazamentos de dia inteiro no nó N11;
- 50 simulações com vazamentos de meio dia no nó N11;

e assim por diante, para todos os nós de vazamento, completando um total de 600 dias de simulação. Ao término, os dados são armazenados em formato CSV.

Para aumentar a diversidade e o volume de dados da mesma RDA, foram realizadas 5 simulações com diferentes sementes de aleatoriedade, resultando em 5 arquivos distintos a serem utilizados no treinamento da GNN.

No processo de conversão da rede hidráulica para o grafo, ocorre uma mudança na nomenclatura, pois os nós do grafo passam a ser numerados a partir de zero. Dessa forma, nos conjuntos de dados, os rótulos de cada classe são atribuídos conforme o número correspondente a cada nó no grafo. A Figura 12 apresenta o grafo equivalente à RDA, e a Tabela 2 detalha a definição dos rótulos.

Figura 12 – Grafo equivalente à RDA do primeiro cenário.



Fonte: Adaptado de Rolle et al. (2022)

Tabela 2 – Atribuições de rótulos para as classes de vazamentos do cenário 1.

| <b>Classe</b>    | <b>Nó no mapa</b> | <b>Atribuição<br/>equivalente no grafo</b> |
|------------------|-------------------|--------------------------------------------|
| Sem vazamento    | M                 | 0                                          |
| Vazamento em N10 | J1                | 1                                          |
| Vazamento em N11 | UC1               | 4                                          |
| Vazamento em N12 | UC2               | 5                                          |
| Vazamento em N13 | UC3               | 6                                          |
| Vazamento em N14 | UC4               | 7                                          |

Fonte: Rolle et al. (2022)

Como a saída do algoritmo indica um nó (neste caso, o mais próximo do vazamento), cada ponto de vazamento é associado ao nó imediatamente anterior. Na ausência de vazamentos, a rede aponta para o nó de medição da entrada (M). Por exemplo, um vazamento em N10 é atribuído ao nó J1, e o mesmo ocorre para os demais pontos. Aos nós 2 e 3 do grafo, não é atribuído nenhum vazamento durante a simulação, considerando que não há vazamentos próximos a eles.

A partir da RDA da Figura 11 e do grafo apresentado na Figura 12, é possível gerar tanto a matriz de adjacência binária, Tabela 3, quanto a matriz de distâncias, Tabela 4. Ambas serão utilizadas nas simulações, alternando-se entre uma e outra, com o objetivo de analisar o impacto de sua utilização no desempenho do modelo.

Tabela 3 – Matriz de adjacência binária do Cenário 1.

|     | M | J1 | J2 | J3 | UC1 | UC2 | UC3 | UC4 |
|-----|---|----|----|----|-----|-----|-----|-----|
| M   | 0 | 1  | 0  | 0  | 0   | 0   | 0   | 0   |
| J1  | 1 | 0  | 1  | 1  | 0   | 0   | 0   | 0   |
| J2  | 0 | 1  | 0  | 0  | 0   | 0   | 1   | 1   |
| J3  | 0 | 1  | 0  | 0  | 1   | 1   | 0   | 0   |
| UC1 | 0 | 0  | 0  | 1  | 0   | 0   | 0   | 0   |
| UC2 | 0 | 0  | 0  | 1  | 0   | 0   | 0   | 0   |
| UC3 | 0 | 0  | 1  | 0  | 0   | 0   | 0   | 0   |
| UC4 | 0 | 0  | 1  | 0  | 0   | 0   | 0   | 0   |

Fonte: Autoria própria.

Tabela 4 – Matriz de distâncias do Cenário 1.

|     | M  | J1 | J2 | J3 | UC1 | UC2 | UC3 | UC4 |
|-----|----|----|----|----|-----|-----|-----|-----|
| M   | 0  | 10 | 0  | 0  | 0   | 0   | 0   | 0   |
| J1  | 10 | 0  | 5  | 5  | 0   | 0   | 0   | 0   |
| J2  | 0  | 5  | 0  | 0  | 0   | 0   | 10  | 10  |
| J3  | 0  | 5  | 0  | 0  | 10  | 10  | 0   | 0   |
| UC1 | 0  | 0  | 0  | 10 | 0   | 0   | 0   | 0   |
| UC2 | 0  | 0  | 0  | 10 | 0   | 0   | 0   | 0   |
| UC3 | 0  | 0  | 10 | 0  | 0   | 0   | 0   | 0   |
| UC4 | 0  | 0  | 10 | 0  | 0   | 0   | 0   | 0   |

Fonte: Autoria própria.

Os hiperparâmetros utilizados para configurar os modelos de GNN foram:

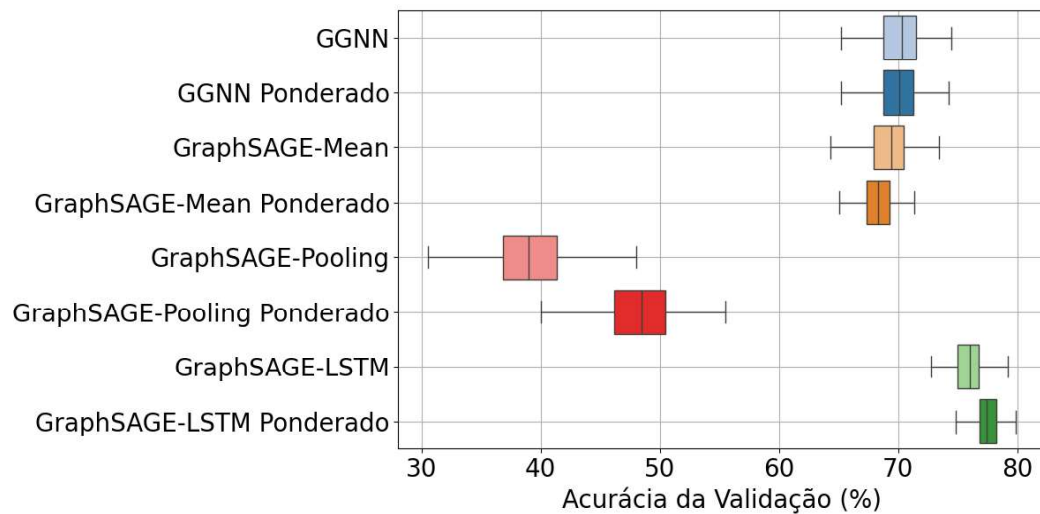
- Número de vizinhos: 3 (Para o algoritmo GraphSAGE);
- Tamanho da vizinhança: 5;
- Tamanho da camada oculta: 256;
- Tamanho do lote: 300;
- Número de subdivisões do algoritmo de validação cruzada (*K-fold*): 10;
- Função de custo (*loss*): entropia cruzada (*Cross-Entropy*);
- Função de otimização: Adam;
- Número de épocas: 300.

Para a implementação do modelo GraphSAGE, foi utilizado o Algoritmo 2, por ser aquele que incorpora todas as características do modelo, realizando a amostragem dos vizinhos antes do processo de agregação.

### 5.2.1. Performance dos modelos para o Cenário 1

A Figura 13 ilustra, por meio de Boxplots, a distribuição da acurácia de validação dos modelos GGNN e GraphSAGE (com agregadores *Mean*, *Pooling* e *LSTM*), comparando a utilização de matriz de adjacência binária com a matriz ponderada por distâncias, no Cenário 1.

Figura 13 - Distribuição da acurácia de validação dos modelos GGNN e GraphSAGE (Mean, Pooling e LSTM) com matriz de adjacência binária e matriz ponderada por distâncias, para o Cenário 1.



Fonte: Autoria própria

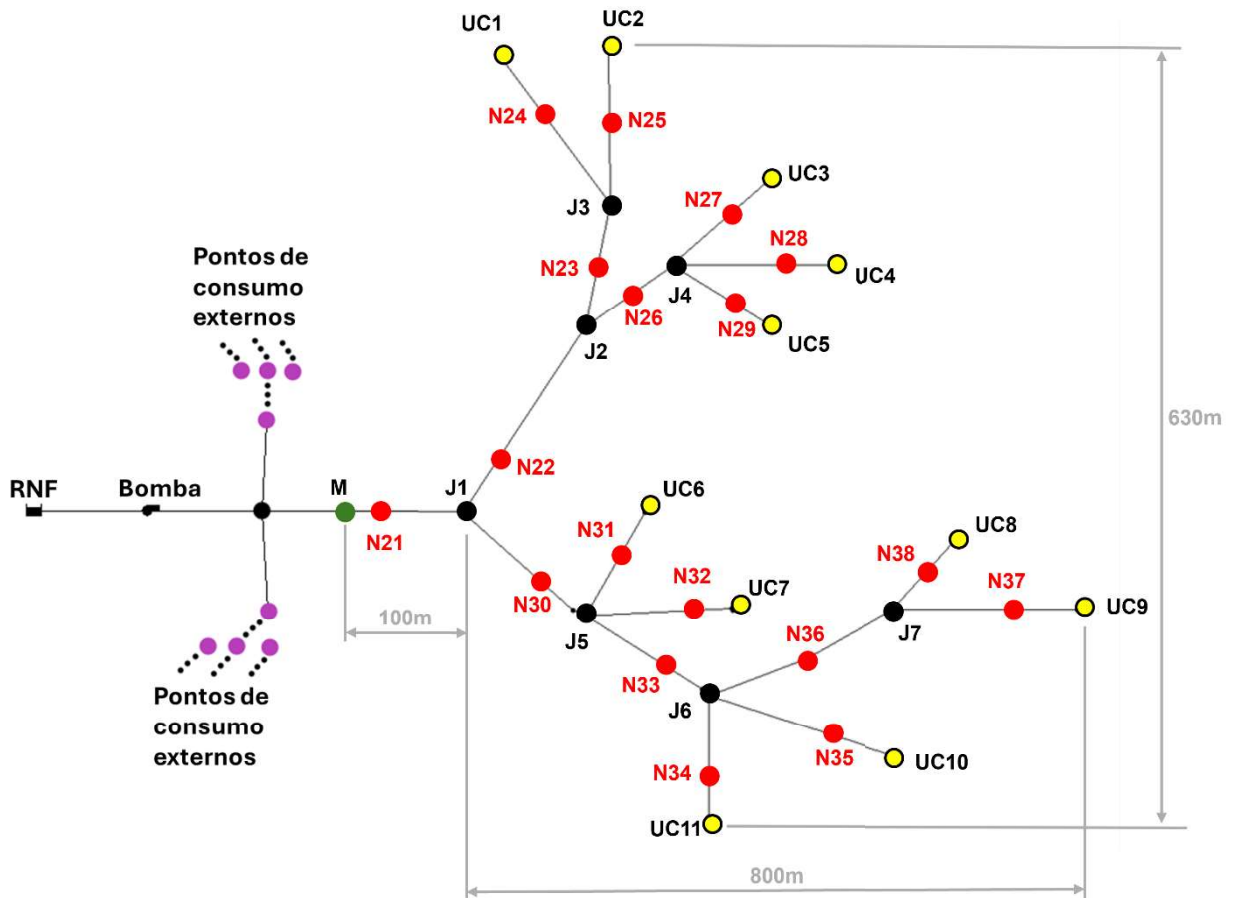
Ao avaliar qualitativamente cada modelo, observa-se que o GGNN ponderado apresentou desempenho muito semelhante ao GGNN não ponderado, com leve redução na mediana da acurácia e discreta diminuição na variabilidade dos resultados, o que sugere que o uso da matriz de distâncias, nesse caso, não trouxe benefícios significativos para o cenário em questão. Para o GraphSAGE-Mean, a introdução da ponderação resultou em desempenho inferior em relação à sua versão não ponderada, embora tenha proporcionado uma ligeira redução da variabilidade. Já o GraphSAGE-Pooling ponderado apresentou um ganho significativo em relação à sua versão não ponderada; no entanto, ambos mantêm desempenho inferior aos demais modelos, indicando que, mesmo com a matriz de ponderação baseada em distâncias, esse agregador não obteve resultados satisfatórios. Por fim, o GraphSAGE-LSTM ponderado apresentou pequena melhora em relação ao modelo não ponderado, com leve redução na dispersão dos resultados, o que pode indicar maior estabilidade, embora o ganho em acurácia tenha sido modesto. Em síntese, a introdução da matriz de ponderação por distâncias promoveu algumas melhorias, mas não de forma significativa ou consistente para todos os modelos, sendo o Pooling o agregador que mais se beneficiou neste cenário.

### 5.3. Cenário 2

A segunda rede, ilustrada na Figura 14, representa um conjunto residencial com 11 casas (marcadas em amarelo), cada uma equipada com um medidor na entrada. A rede também possui um medidor inteligente instalado na entrada principal da tubulação (marcado em verde). Foram incluídos 30 pontos de consumo adicionais (em roxo) para simular demandas variadas dos demais consumidores urbanos. Os pontos pretos indicam junções, enquanto os pontos vermelhos representam locais de possíveis vazamentos. As distâncias de referência estão destacadas por cotas, fora de escala.

Este modelo também conta com 16 padrões diários de consumo, aleatoriamente atribuídos às unidades consumidoras para as simulações. Outros 11 padrões adicionais modelam o consumo nos pontos externos e são distribuídos por sorteio. Para esta rede, os vazamentos também foram configurados de tal maneira que apresentaram uma vazão média de 3 LPM. No entanto, aqui eles foram configurados para ocorrer por 24 horas, ou seja, durante o dia todo.

Figura 14 - RDA do segundo cenário.



Fonte: Adaptado de Rolle et al. (2024)

A sequência de simulações é dada por:

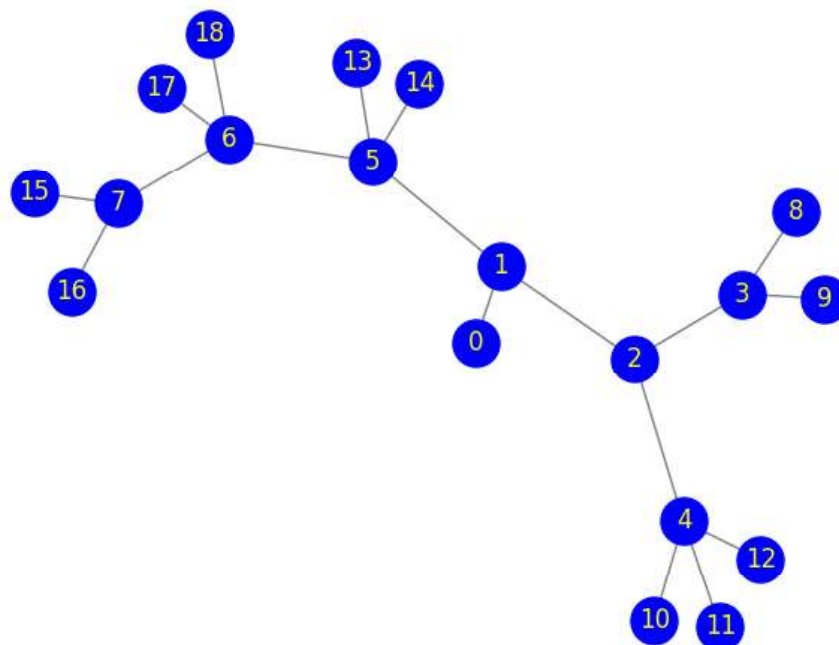
- 100 simulações sem vazamento;
- 100 simulações com vazamentos de dia inteiro no nó N21;
- 100 simulações com vazamentos de dia inteiro no nó N22;
- 100 simulações com vazamentos de dia inteiro no nó N23;

e assim por diante, para todos os nós de vazamento, completando um total de 1900 dias de simulação. Ao término, os dados são armazenados em formato CSV.

Para esta RDA, também foram realizadas cinco simulações com diferentes sementes de aleatoriedade, a fim de gerar maior diversidade de dados. Cada simulação resultou em um arquivo distinto utilizado no treinamento da GNN.

Assim como ocorre no primeiro cenário, no processo de conversão da rede hidráulica para o grafo, acontece uma mudança na nomenclatura. A Figura 15 apresenta o grafo equivalente à RDA, e a Tabela 5 detalha a definição dos rótulos.

Figura 15 - Grafo equivalente à RDA do segundo cenário.



Fonte: Adaptado de Rolle et al. (2024)

Tabela 5 - Atribuições de rótulos para as classes de vazamentos do cenário 2.

| Classe           | Nó no mapa | Atribuição<br>equivalente no grafo |
|------------------|------------|------------------------------------|
| Sem vazamento    | M          | 0                                  |
| Vazamento em N21 | J1         | 1                                  |
| Vazamento em N22 | J2         | 2                                  |
| Vazamento em N23 | J3         | 3                                  |
| Vazamento em N24 | UC1        | 8                                  |
| Vazamento em N25 | UC2        | 9                                  |
| Vazamento em N26 | J4         | 4                                  |
| Vazamento em N27 | UC3        | 10                                 |
| Vazamento em N28 | UC4        | 11                                 |
| Vazamento em N29 | UC5        | 12                                 |
| Vazamento em N30 | J5         | 5                                  |
| Vazamento em N31 | UC6        | 13                                 |
| Vazamento em N32 | UC7        | 14                                 |
| Vazamento em N33 | J6         | 6                                  |
| Vazamento em N34 | UC11       | 18                                 |
| Vazamento em N35 | UC10       | 17                                 |
| Vazamento em N36 | J7         | 7                                  |
| Vazamento em N37 | UC9        | 16                                 |
| Vazamento em N38 | UC8        | 15                                 |

Fonte: Rolle et al. (2022)

Cada ponto de vazamento é vinculado ao nó imediatamente anterior a ele. Na ausência de vazamentos, a rede direciona a saída para o nó de medição da entrada (M). Por exemplo, o vazamento em N21 é atribuído ao nó J1, e assim sucessivamente para os demais pontos.

A partir da RDA apresentada na Figura 14 e do grafo correspondente da Figura 15, é possível gerar tanto a matriz de adjacência (Tabela 8) quanto a matriz de distâncias (Tabela 9), referentes ao Cenário 2, ambas disponíveis no Apêndice A desta dissertação. Essas matrizes também serão utilizadas nas simulações, sendo alternadas com o objetivo de analisar o impacto de sua utilização no desempenho do modelo.

Os hiperparâmetros utilizados para configurar os modelos de GNN foram:

- Número de vizinhos: 3 (Para o algoritmo GraphSAGE);
- Tamanho da vizinhança: 5;
- Tamanho da camada oculta: 256;
- Tamanho do lote: 300;
- Número de subdivisões do algoritmo de validação cruzada (*K-fold*): 10;
- Função de custo (*loss*): entropia cruzada (*Cross-Entropy*);

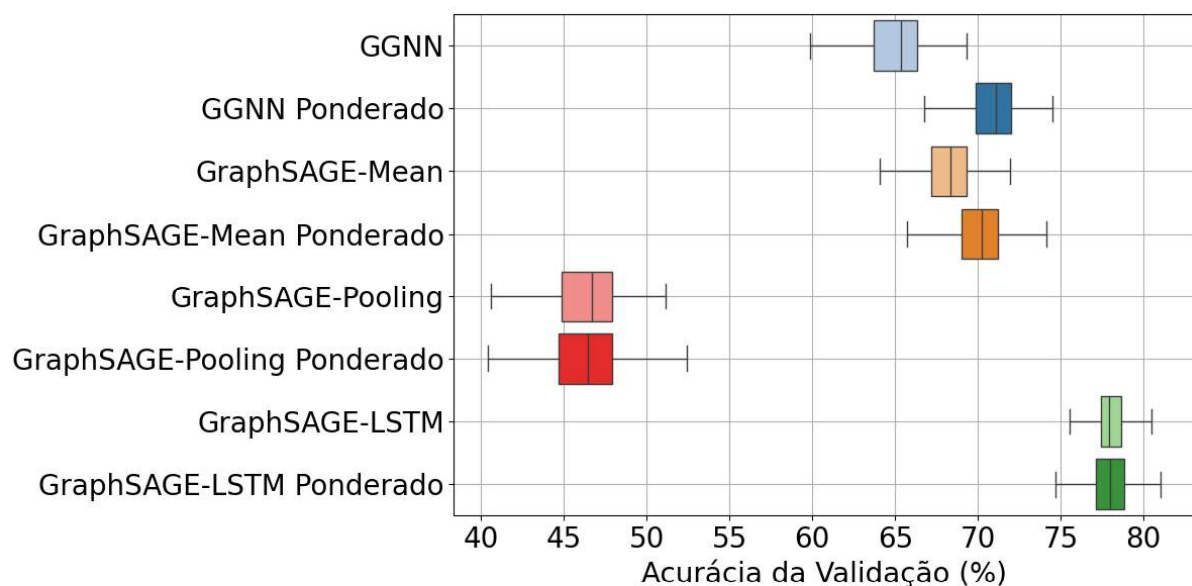
- Função de otimização: Adam;
- Número de épocas: 200.

Neste caso, também foi implementado o Algoritmo 2, pela mesma razão mencionada anteriormente.

### 5.3.1. Performance dos modelos para o Cenário 2

A Figura 16 ilustra, por meio de Boxplots, a distribuição das acurácias da validação obtidas para cada configuração do modelo e tipo de matriz utilizada, para o Cenário 2.

Figura 16 - Distribuição da acurácia de validação dos modelos GGNN e GraphSAGE (*Mean*, *Pooling* e *LSTM*) com matriz de adjacência binária e matriz ponderada por distâncias, para o Cenário 2.



Fonte: Autoria própria

A análise qualitativa deste cenário evidencia diferenças relevantes no comportamento dos modelos com e sem ponderação por distância. O GGNN apresentou desempenho superior em sua versão ponderada, com valores de acurácia mais elevados e distribuição mais concentrada, indicando melhora no desempenho médio. No GraphSAGE-*Mean*, a ponderação também resultou em ganho de acurácia, sugerindo benefício da incorporação da informação de distância entre os nós. Para o GraphSAGE-*Pooling*, ambos os modelos apresentaram desempenhos baixos e semelhantes, com grande variabilidade e sobreposição nas faixas interquartis, indicando que esse agregador não foi eficaz, independentemente da matriz

utilizada. Já o GraphSAGE-LSTM se destacou entre todos os modelos, apresentando as maiores acurácias. A versão ponderada teve leve superioridade em relação à não ponderada, mas ambas mantiveram desempenho elevado e consistente, com distribuição compacta. Esses resultados sugerem que a matriz de ponderação baseada em distâncias tende a beneficiar modelos mais robustos, enquanto em arquiteturas menos eficazes, como o *Pooling*, seu impacto é mínimo.

#### 5.4. Considerações sobre os resultados

As Tabela 6 e 7 apresentam o resumo dos resultados obtidos para as três variações de agregadores utilizadas no modelo GraphSAGE, juntamente com o desempenho do modelo GGNN, que serviu como *benchmark* para este trabalho.

Tabela 6 – Apresentação das métricas de avaliação dos modelos de GNNs aplicados na tarefa para o Cenário 1.

| Modelo                      | Cenário 1    |              |              |              |
|-----------------------------|--------------|--------------|--------------|--------------|
|                             | Acurácia (%) | Precisão(%)  | Recall(%)    | F1-Score(%)  |
| GGNN                        | 69,64        | 66,87        | 67,43        | 67,15        |
| GGNN Ponderado              | 69,39        | 66,06        | 66,44        | 66,25        |
| GraphSAGE-Mean              | 68,58        | 66,92        | 67,16        | 67,04        |
| GraphSAGE-Mean Ponderado    | 67,70        | 64,74        | 65,12        | 64,93        |
| GraphSAGE-Pooling           | 38,85        | 29,33        | 25,72        | 27,41        |
| GraphSAGE-Pooling Ponderado | 47,94        | 35,06        | 32,20        | 33,67        |
| GraphSAGE-LSTM              | 75,76        | 70,80        | 71,22        | 71,01        |
| GraphSAGE-LSTM Ponderado    | <b>77,49</b> | <b>71,66</b> | <b>73,14</b> | <b>72,39</b> |

Fonte: Autoria própria.

Tabela 7 - Apresentação das métricas de avaliação dos modelos de GNNs aplicados na tarefa para o Cenário 2.

| Modelo                      | Cenário 2    |              |              |              |
|-----------------------------|--------------|--------------|--------------|--------------|
|                             | Acurácia (%) | Precisão(%)  | Recall(%)    | F1-Score(%)  |
| GGNN                        | 64,42        | 64,70        | 60,63        | 62,60        |
| GGNN Ponderado              | 70,49        | 70,33        | 69,84        | 70,08        |
| GraphSAGE-Mean              | 67,40        | 66,81        | 63,77        | 65,25        |
| GraphSAGE-Mean Ponderado    | 69,23        | 69,35        | 68,47        | 68,91        |
| GraphSAGE-Pooling           | 46,40        | 54,30        | 43,19        | 48,11        |
| GraphSAGE-Pooling Ponderado | 46,30        | 54,67        | 45,50        | 49,66        |
| GraphSAGE-LSTM              | <b>78,01</b> | 77,66        | <b>77,09</b> | <b>77,37</b> |
| GraphSAGE-LSTM Ponderado    | 77,90        | <b>77,73</b> | 76,71        | 77,22        |

Fonte: Autoria própria.

Nos dois cenários analisados, observa-se uma superioridade consistente dos modelos baseados no agregador GraphSAGE-LSTM, tanto com matriz binária quanto ponderada, em relação às demais abordagens avaliadas. Essa superioridade está associada à capacidade do agregador LSTM de capturar padrões complexos e relações não lineares entre os nós, utilizando a estrutura sequencial de vizinhança para gerar *embeddings* mais expressivos.

No Cenário 1, o modelo GraphSAGE-LSTM Ponderado obteve os melhores resultados em todas as métricas, com 77,49% de acurácia, 71,66% de precisão, 73,14% de recall e 72,39% de F1-Score, indicando uma performance significativamente superior em relação ao *benchmark* GGNN, cujo melhor desempenho (sem ponderação) foi de 69,64% de acurácia e 67,15% de F1-Score. Esse resultado pode ser explicado pelo fato de que a arquitetura LSTM consegue integrar as informações de vizinhança preservando dependências mais longas, enquanto o GGNN, apesar de usar um mecanismo recorrente, tende a ter maior dificuldade em capturar variações mais sutis nos dados.

Por outro lado, os modelos com agregador *Pooling*, especialmente sem ponderação, apresentaram desempenho inferior, com destaque para o GraphSAGE-*Pooling*, que obteve apenas 38,85% de acurácia e 27,41% de F1-Score, o pior desempenho geral. Tal resultado sugere que esse tipo de agregador, que realiza operações de máximo (para o caso desse experimento) no conjunto de vizinhos, perde informação detalhada das interações locais, o que

dificultou o aprendizado dos padrões necessários para a localização de vazamentos neste cenário.

Para este cenário, os resultados evidenciam que o uso da matriz de ponderação baseada nas distâncias físicas entre os nós teve um impacto positivo, principalmente nos modelos com agregadores LSTM e *Pooling*, que aumentaram seu desempenho de acurácia e F1-Score. Já para os modelos GGNN e GraphSAGE-Mean, por mais que sua utilização não tenha elevado seus desempenhos, ela serviu, mesmo que sutil, para reduzir a dispersão dos resultados. Essa ponderação permitiu que o modelo incorporasse informações estruturais mais realistas da rede, atribuindo peso às conexões, trazendo melhora para a agregação dos dados. A diferença de desempenho entre o GraphSAGE-LSTM Ponderado e sua versão não ponderada foi de aproximadamente 1,7 pontos percentuais em F1-Score, contribuindo, mesmo que simbolicamente, com a hipótese  $H_{12}$  sobre a eficácia do uso da matriz de distâncias com ponderação.

No Cenário 2, a tendência observada se manteve. O modelo GraphSAGE-LSTM, mesmo sem ponderação, obteve os melhores resultados, com 78,01% de acurácia, 77,66% de precisão, 77,09% de recall e 77,37% de F1-Score. Quando associada à matriz de distâncias, a versão ponderada manteve desempenho elevado, com 77,90% de acurácia e 77,22% de F1-Score, confirmando a robustez do modelo mesmo em cenários distintos. A arquitetura LSTM, ao capturar dependências mais complexas entre nós, manteve-se menos sensível à variação da matriz.

Observou-se que tanto o modelo GGNN quanto o GraphSAGE-Mean apresentaram ganhos de desempenho ao utilizarem a matriz de ponderação neste cenário, sendo que o modelo GGNN apresentou a variação mais expressiva, com a acurácia elevando-se de 64,42% para 70,49%, correspondendo a um aumento de 6,07 pontos percentuais. Esse resultado evidencia que a incorporação de informações físicas da rede, por meio da ponderação das conexões entre os nós, enriquece a representação estrutural dos dados e potencializa a capacidade preditiva dos modelos. Esses achados podem contribuir com a hipótese  $H_{12}$ , ajundando a demonstrar a relevância do uso da ponderação física na modelagem das relações topológicas da rede.

Ainda no Cenário 2, os modelos baseados no agregador *Pooling* voltaram a apresentar desempenho significativamente inferior, para ambos os casos, com F1-Scores abaixo de 50%, o que confirma a limitação observada anteriormente para essa estratégia de agregação.

Em geral, as observações detalhadas em ambos os cenários revelaram informações relevantes sobre a eficácia de diferentes abordagens para a detecção e localização de vazamentos em RDAs modeladas como grafos. A análise dos resultados suporta a rejeição da

hipótese nula  $H_{01}$ , que afirma que o uso do algoritmo GraphSAGE, com diferentes agregadores, não apresenta melhora o desempenho de detecção e localização de vazamentos em RDAs, quando comparado aos algoritmos de GNNs utilizados anteriormente, e sustenta a aceitação da hipótese alternativa  $H_{11}$ , onde é afirmado o contrário. Contudo, é fundamental ressaltar que a superioridade do algoritmo GraphSAGE não se aplica de maneira homogênea a todos os seus agregadores, sendo o GraphSAGE-LSTM o principal responsável pela melhoria de desempenho em relação ao *benchmark* GGNN.

Quanto à segunda pergunta de pesquisa, os resultados permitem, rejeitar a hipótese nula  $H_{02}$  e aceitar a hipótese alternativa  $H_{12}$ , que afirma que o uso de uma matriz de ponderação baseada nas distâncias físicas melhora o desempenho dos modelos GNNs. A ponderação incorporou informações estruturais mais realistas da rede, atribuindo pesos às conexões, o que, por sua vez, melhorou a agregação dos dados, possibilitando melhora no desempenho dos modelos e diminuição da variabilidade dos resultados.

## 6. CONCLUSÃO

Esta pesquisa teve como objetivo principal investigar a aplicação do algoritmo GraphSAGE na tarefa de detecção e localização de vazamentos em RDAs, analisando seu desempenho em comparação com abordagens baseadas em GNNs previamente utilizadas, como a GGNN. Adicionalmente, buscou-se avaliar o impacto da substituição da matriz de adjacência binária por uma matriz de ponderação baseada nas distâncias físicas entre os nós da rede. As investigações foram conduzidas por meio da formulação de duas perguntas de pesquisa: (P1) se a aplicação do GraphSAGE melhora a acurácia na detecção e localização de vazamentos; e (P2) se a utilização de uma matriz de ponderação baseada em distâncias físicas melhora o desempenho do modelo GNN em comparação com a matriz binária.

Em relação à primeira pergunta de pesquisa (P1), os resultados obtidos nos cenários simulados indicaram que o algoritmo GraphSAGE, quando comparado à GGNN, apresentou melhorias nas métricas de desempenho, especialmente em termos de acurácia e F1-score. Esses ganhos foram observados com maior consistência quando utilizado o agregador LSTM. Diante disso, conclui-se que a hipótese nula  $H_{01}$  foi rejeitada e a hipótese alternativa  $H_{11}$  foi confirmada, ou seja, o uso do GraphSAGE melhora a acurácia da detecção e localização de vazamentos em RDAs, superando o algoritmo utilizado como referência.

Quanto à segunda pergunta de pesquisa (P2), verificou-se que a utilização da matriz de ponderação baseada nas distâncias físicas entre os nós também levou a melhorias no desempenho dos modelos em comparação à matriz de adjacência binária. Em ambos os cenários, a introdução dessa matriz resultou em ganhos em acurácia e estabilidade do modelo, com destaque para o Cenário 2, onde verificou-se maior sensibilidade dos modelos quanto à qualidade da representação estrutural da rede. Dessa forma, a hipótese nula  $H_{02}$  foi rejeitada e a hipótese alternativa  $H_{12}$  foi confirmada, evidenciando que a matriz baseada em distâncias contribui para o aprendizado do modelo e para a representação mais fiel das propriedades físicas da rede.

Portanto, com base nas evidências empíricas apresentadas, pode-se afirmar que ambas as contribuições propostas por esta dissertação, o uso do GraphSAGE com diferentes estratégias de agregação e a substituição da matriz binária por uma matriz ponderada por distâncias, resultaram em melhorias relevantes na tarefa de detecção e localização de vazamentos em RDAs. Os resultados obtidos reforçam o potencial da abordagem proposta para aplicações práticas no setor de saneamento, contribuindo para o desenvolvimento de sistemas inteligentes, eficientes e sustentáveis de monitoramento de perdas hídricas.

Como trabalhos futuros, sugerem-se investigações que ampliem a complexidade e o realismo dos cenários avaliados, contemplando: a incorporação da variável altura ao modelo hidráulico; a adição de ruídos aos sinais de pressão para simular leituras reais de sensores; estudos sobre a quantidade mínima e o posicionamento ótimo de sensores na rede; a aplicação do método em redes com topologia malhada ou mista; a exploração de diferentes formas de ponderação das relações entre os nós; e o uso de arquiteturas mais avançadas de GNN, especialmente aquelas baseadas em mecanismos de atenção, capazes de capturar padrões mais complexos e heterogêneos.

## REFERÊNCIAS

- Abdulshaheed, A.; Mustapha, F.; Ghavamian, A. A pressure-based method for monitoring leaks in a pipe distribution system: A review. *Renewable and Sustainable Energy Reviews*, Elsevier, v. 69, p. 902–911, 2017.
- ABOU RIDA, A.; AMHAZ, R.; PARREND, P. Anomaly detection for cybersecurity using convolutional graph neural network compared to graph analysis and graph embedding. In: *International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021. Proceedings [...]. Las Vegas: IEEE, 2021. p. 1234-1240.
- Adedeji, K. B.; Hamam, Y.; Abe, B. T.; Abu-Mahfouz, A. M. Towards achieving a reliable leakage detection and localization algorithm for application in water piping networks: An overview. *IEEE Access*, IEEE, v. 5, p. 20272–20285, 2017.
- Ayati, A. H.; Haghighi, A.; Lee, P. Statistical review of major standpoints in hydraulic transient-based leak detection. *Journal of Hydraulic Structures*, Shahid Chamran University of Ahvaz, v. 5, n. 1, p. 1–26, 2019.
- Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez- Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261.
- Bermúdez, J.-R.; López-Estrada, F.-R.; Besançon, G.; Valencia-Palomo, G.; Torres, L.; Hernández, H.-R. Modeling and simulation of a hydraulic network for leak diagnosis. *Mathematical and computational applications*, Multidisciplinary Digital Publishing Institute, v. 23, n. 4, p. 70, 2018.
- Candelieri, A., Conti, D., and Archetti, F. (2014). A graph based analysis of leak localization in urban water networks. *Procedia Engineering*, 70, 228-237. doi:10.1016/j.proeng.2014.02.026. 12th International Conference on Computing and Control for theWater Industry, CCWI2013.
- Chan, T. K.; Chin, C. S.; Zhong, X. Review of current technologies and proposed intelligent methodologies for water distributed network leakage detection. *IEEE Access*, IEEE, v. 6, p. 78846–78867, 2018.
- Chen, C.; Li, Y.; Jiang, M. An improved GraphSAGE to detect power system anomaly based on time-neighbor feature. *Energy Reports*, v. 9, p. 930-937, 2023. DOI: <https://doi.org/10.1016/j.egyr.2022.12.055>.
- 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.

Eliades, D. G.; Kyriakou, M.; Vrachimis, S.; Polycarpou, M. M. 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.

Fan, X., Zhang, X. & Yu, X.(B. Machine learning model and strategy for fast and accurate detection of leaks in water supply network. *J Infrastruct Preserv Resil* 2, 10 (2021). <https://doi.org/10.1186/s43065-021-00021-6>.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017, July). Neural message passing for quantum chemistry. In *International conference on machine learning* (pp. 1263-1272). PMLR.

Hamilton, W.L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. In NIPS. doi:10.48550/arXiv:1706.02216[cs.SI].

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780.

Javadiha, M.; Blesa, J.; Soldevila, A.; Puig, V. Leak localization in water distribution networks using deep learning. In: IEEE. *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*. [S.l.], 2019. p. 1426–1431.

Kang, J., Park, Y.J., Lee, J., Wang, S.H., and Eom, D.S. (2018). Novel leakage detection by ensemble CNN-SVM and graph-based localization in water distribution systems. *IEEE Transactions on Industrial Electronics*, 65(5), 4279–4289. doi:10.1109/TIE.2017.2764861.

Kipf, T.N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. doi:10.48550/arXiv.1609.02907.

Li, C.T., Tsai, Y.C., and Liao, J.C. (2023). Graph neural networks for tabular data learning. In 2023 IEEE 39<sup>th</sup> International Conference on Data Engineering (ICDE), 3589–3592. doi:10.1109/ICDE55515.2023.00275.

Liang, B.; Verma, S.; Xu, J.; Liang, S.; Li, Z.; Wang, Y.; Chen, F. A data driven approach for leak detection with smart sensors. In: IEEE. *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. [S.l.], 2020. p. 1311–1316.

Loureiro, D.; Amado, C.; Martins, A.; Vitorino, D.; Mamade, A.; Coelho, S. T. Water distribution systems flow monitoring and anomalous event detection: A practical approach. *Urban Water Journal*, Taylor & Francis, v. 13, n. 3, p. 242–252, 2016.

Lučin, I.; Lučin, B.; Čarija, Z.; Sikirica, A. Data-Driven Leak Localization in Urban Water Distribution Networks Using Big Data for Random Forest Classifier. *Mathematics* 2021, 9, 672.

Marfo, W.; Lazarova-Molnar, S.; Al-Shemmery, H. Enhancing network anomaly detection using graph neural networks. In: *Mediterranean Communication and Computer Networking Conference (MedComNet)*, 22., 2024, Roma. Proceedings [...]. Roma: IEEE, 2024. p. 1-7.

Medina, M. M. G. *Detecção de vazamentos e alterações em redes de distribuição de água para abastecimento, durante a operação, usando sinais de pressão*. Tese (Doutorado) — Universidade de São Paulo, 2017.

Mpiana, L.A., Hamam, Y., and Abu-Mahfouz, A.M. (2017). Graph based hydraulic modelling of pressure in water distribution networks. In 2017 IEEE AFRICON, 1558-1563. doi:10.1109/AFRICON.2017.8095714.

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*, Multidisciplinary Digital Publishing Institute, v. 20, n. 18, p. 5204, 2020.

Oliveira, E.; Fonseca, M.; Kappes, D.; Medeiros, A.; Stefanini, I. Leak detection system using machine learning techniques. In: *6th International Congress on Automation in Mining, At Santiago, Chile*. [S.l.: s.n.], 2018.

Pérez-Pérez, E.; López-Estrada, F. R.; Valencia-Palomo, O, G.; Torres,L.; PUIG, V.; Mina-Antonio, J. Leak diagnosis in pipelines using a combined artificial neural network approach. *Control Engineering Practice*, Elsevier, v. 107, p. 104677, 2021.

Perez, R.; Sanz, G.; Puig, V.; Quevedo, J.; Escofet, M. A. C.; Nejari, F.; Meseguer, J.; Cembrano, G.; Tur, J. M. M.; Sarrate, R.; Leak localization in water networks: A model-based methodology using pressure sensors applied to a real network in barcelona [applications of control]. *IEEE control systems magazine*, IEEE, v. 34, n. 4, p. 24–36, 2014.

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*, Multidisciplinary Digital Publishing Institute, v. 11, n. 2, p. 518, 2019.

Rolle R.P., *Detecção e Localização de Vazamentos utilizando Internet das Coisas e Aprendizado de Máquina*. Tese (Doutorado) – Universidade Estadual Paulista “Julio de Mesquita Filho”, 2024.

Rolle R.P., Monteiro, L. N., Tomazini, L. R., Monteiro, L. N. and Godoy, E. P., “Detecção de vazamentos em redes hidráulicas utilizando Redes Neurais de Grafos”, (2022). Congresso Brasileiro de Automática - CBA. 3. 10.20906/CBA2022/3688.

Rolle, R. P., Rodrigues, W. C., Tomazini, L. R., Monteiro, L. N. and Godoy, E. P., "Leveraging graph-based leak localization in water distribution networks," 2024 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0 & IoT), Firenze, Italy, 2024, pp. 192-197, doi: 10.1109/MetroInd4.0IoT61288.2024.10584129.

- Romano, M.; Kapelan, Z.; SAVIĆ, D. A. Automated detection of pipe bursts and other events in water distribution systems. *Journal of Water Resources Planning and Management*, American Society of Civil Engineers, v. 140, n. 4, p. 457–467, 2014.
- Rossman, L. A.; Woo, H.; TrybyRYBY, M.; Shang, F.; Janke, R.; Haxston, T. EPANET users manual. EPA-Environmental Protection Agency, 1994.
- Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1), 61-80. doi:10.1109/TNN.2008.2005605.
- Scott, S. L.; Barrufet, M. A., *Worldwide assessment of industry leak detection capabilities for single & multiphase pipelines*. [S.l.]: Offshore Technology Research Center College Station, 2003.
- SILVA, C. O. d. *Modelagem de rede de distribuição de água com ênfase no controle de perdas*. Dissertação (Mestrado) — Universidade Federal da Paraíba, 2014.
- Sun, C.; Parellada, B.; Puig, V.; Cembrano, G. Leak localization in water distribution networks using pressure and data-driven classifier approach. *Water*, Multidisciplinary Digital Publishing Institute, v. 12, n. 1, p. 54, 2020.
- 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.R., Rodrigues, W.C., Rolle, R.P., Simões, A.S., Colombini, E.L., and Godoy, E.P. (2024). Graphleak: A realistic dataset to detect and locate leaks in water distribution networks. URL <https://github.com/gasiepgodoy/WDN-Models-and-Data-Sets>.
- Torres, L.; Jiménez-Cabas, J.; González, O.; Molina, L.; López-Estrada, F.-R. Kalman filters for leak diagnosis in pipelines: Brief history and future research. *Journal of Marine Science and Engineering*, Multidisciplinary Digital Publishing Institute, v. 8, n. 3, p. 173, 2020.
- UN-Water. *UN World Water Development Report 2024: Water for Prosperity and Peace*. Paris: UNESCO/UN-Water, 19 mar. 2024.
- Velayudhan, N.K., Pradeep, P., Rao, S.N., Devidas, A.R., and Ramesh, M.V. (2022). Iot-enabled water distribution systems - a comparative technological review. *IEEE Access*, 10, 101042-101070. doi:10.1109/ACCESS.2022.3208142.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.
- Walski, T. M.; Chase, D. V.; Savic D. A.; Grayman, W.; Beckwith, S.; Koelle E. Advanced water distribution modeling and management. Haestad press, 2003.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. and Philip. S. Y., "A Comprehensive Survey on Graph Neural Networks," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4-24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.

Zanfei, A., Menapace, A., Brentan, B.M., Righetti, M., and Herrera, M. (2022). Novel approach for burst detection in water distribution systems based on graph neural networks. *Sustainable Cities and Society*, 86, 104090. doi:<https://doi.org/10.1016/j.scs.2022.104090>.

Zhou, B.; Lau, V.; Wang, X. Machine-learning-based leakage-event identification for smart water supply systems. *IEEE Internet of Things Journal*, IEEE, v. 7, n. 3, p. 2277–2292, 2019.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.; Graph neural networks: A review of methods and applications, *AI Open*, Volume 1, 2020, Pages 57-81, ISSN 2666-6510, doi.org/10.1016/j.aiopen.2021.01.001.

Zhou, X.; Tang, Z.; Xu, W.; Meng, F.; Chu, X.; Xin, K.; Fu, G. Deep learning identifies accurate burst locations in water distribution networks. *Water research*, Elsevier, v. 166, p. 115058, 2019.

## APÊNDICES

Apêndice A – Matriz de adjacência binária e matriz de distâncias do Cenário 2.

Tabela 8 - Matriz de adjacência binária do Cenário 2.

|      | M | J1 | J2 | J3 | J4 | J5 | J6 | J7 | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | UC8 | UC9 | UC10 | UC11 |
|------|---|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|
| M    | 0 | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| J1   | 1 | 0  | 1  | 0  | 0  | 1  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| J2   | 0 | 1  | 0  | 1  | 1  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| J3   | 0 | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| J4   | 0 | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0   | 0   | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0    | 0    |
| J5   | 0 | 1  | 0  | 0  | 0  | 0  | 1  | 0  | 0   | 0   | 0   | 0   | 0   | 1   | 1   | 0   | 0   | 0    | 0    |
| J6   | 0 | 0  | 0  | 0  | 0  | 1  | 0  | 1  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1    | 1    |
| J7   | 0 | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 1   | 0    | 0    |
| UC1  | 0 | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC2  | 0 | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC3  | 0 | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC4  | 0 | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC5  | 0 | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC6  | 0 | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC7  | 0 | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC8  | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC9  | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC10 | 0 | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC11 | 0 | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |

Fonte: Autoria própria.

Tabela 9 - Matriz de ponderação baseada nas distâncias do Cenário 2.

|      | M   | J1  | J2  | J3  | J4  | J5  | J6  | J7  | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | UC8 | UC9 | UC10 | UC11 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|
| M    | 0   | 100 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| J1   | 100 | 0   | 180 | 0   | 0   | 131 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| J2   | 0   | 180 | 0   | 102 | 86  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| J3   | 0   | 0   | 102 | 0   | 0   | 0   | 0   | 0   | 150 | 130 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| J4   | 0   | 0   | 86  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 106 | 130 | 94  | 0   | 0   | 0   | 0   | 0    | 0    |
| J5   | 0   | 131 | 0   | 0   | 0   | 0   | 120 | 0   | 0   | 0   | 0   | 0   | 0   | 98  | 120 | 0   | 0   | 0    | 0    |
| J6   | 0   | 0   | 0   | 0   | 0   | 120 | 0   | 165 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 158  | 100  |
| J7   | 0   | 0   | 0   | 0   | 0   | 0   | 165 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 74  | 150 | 0    | 0    |
| UC1  | 0   | 0   | 0   | 150 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC2  | 0   | 0   | 0   | 130 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC3  | 0   | 0   | 0   | 0   | 106 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC4  | 0   | 0   | 0   | 0   | 130 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC5  | 0   | 0   | 0   | 0   | 94  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC6  | 0   | 0   | 0   | 0   | 0   | 98  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC7  | 0   | 0   | 0   | 0   | 0   | 120 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC8  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 74  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC9  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 150 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC10 | 0   | 0   | 0   | 0   | 0   | 0   | 158 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |
| UC11 | 0   | 0   | 0   | 0   | 0   | 0   | 100 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    |

Fonte: Autoria própria.