

POLYTECHNIC OF TURIN

Master's Degree in Data Science and Engineering



Master's Degree Thesis

Subgraph-Based Training Augmentation for Graph Anomaly Detection with Very Few Labels

Supervisors

Prof. Paolo GARZA

Prof. Hamida SEBA

Candidate

Rino DI CHIARA

July 2026

A tutte le lacrime versate e ai sacrifici fatti.
Ai silenzi, alla rabbia, alla solitudine, alla pazienza, alla costanza.
Al coraggio di andare sempre avanti e alla paura di tornare indietro.
A chi mi ha ferito, deluso, tradito.
Alla mediocrità degli altri e a chi si è battuto per me.
A chi non mi ha mai rinnegato e mi ha sempre scelto.
A quelli che mi vogliono davvero bene.
A mio padre Claudio per essere un padre e uomo di profondo valore.
A mia mamma Laura per essere stata madre portatrice di unità, speranza e fiducia
nel figlio libero.
Alla forza mia sorella Giusy nel restare ed essere testimone solido delle mie origini.
Alla mia famiglia, unica certezza della mia vita.

Abstract

Graph anomaly detection (GAD) aims to identify nodes that behave differently from the rest of the graph. It is an important task for many applications, from fraud detection to network security, and it becomes particularly challenging when only a small number of anomalous nodes are labeled - a common situation in practice, where anomalies are rare by nature and expensive to annotate. While unsupervised methods can achieve good performance without labeled data, they are unable to exploit that limited supervision that is often available in real-world scenarios.

In this work, we study GAD under very few labels and propose a training augmentation framework based on Subgraph Forest Fire Sampling (SFFS). Given a small set of labeled anomaly seeds, SFFS builds local subgraphs by expanding from each seed to its surrounding neighborhood. The nodes discovered in these subgraphs are added to the training set, enriching the structural context available to the detector without modifying node features or introducing additional architectural components. A GraphSAGE classifier is then trained on the augmented training set.

We evaluate the proposed framework on six benchmark datasets - Cora, Citeseer, BlogCatalog, Flickr, ACM, and Pubmed - with labeled ratios ranging from 1% to 10% of known anomalies. Results show that SFFS augmentation consistently improves detection performance. The gains are particularly evident in AUC-PR, with improvements up to +0.183 on Pubmed and +0.157 on Cora at 10% labeled ratio, while AUC-ROC improvements are more moderate and dataset-dependent. The improvements are more pronounced at higher supervision levels and vary across datasets with different structural characteristics. This indicates that the method benefits from the presence of informative local neighborhoods around anomaly seeds, whose quality and density depend on the graph structure.

Table of Contents

List of Tables	IV
List of Figures	V
1 Introduction	1
1.1 Objectives	2
1.2 Thesis Structure	3
2 Background	4
2.1 Graph Fundamentals	4
2.2 Graph Neural Networks	5
2.2.1 The Message Passing Framework	5
2.2.2 Over-smoothing and its Implications	6
2.2.3 Graph Convolutional Networks	7
2.2.4 Other GNN Variants	8
2.3 Graph Anomaly Detection	11
2.3.1 Anomalies in Graphs	11
2.3.2 Problem Formulation	12
2.3.3 Detection Approaches	13
2.3.4 Evaluation Metrics	14
3 Related Work	17
3.1 Overview of the Literature	17
3.1.1 Selection Methodology	17
3.1.2 Key Trends and Statistics	18
3.2 Learning techniques	19
3.2.1 Contrastive Learning	19
3.2.2 Meta-Learning	20
3.2.3 Prompt Tuning	20
3.2.4 Data-Centric Learning	21
3.2.5 Generative, Reinforcement, and Knowledge Distillation	22

3.3	Problem Settings and Experimental Issues	22
3.4	Positioning of this Work	23
4	Proposed Method	27
4.1	Overview	27
4.2	Subgraph Construction via SFFS	28
4.2.1	Forest Fire Sampling Mechanism	28
4.2.2	Subgraph Construction Strategies	31
5	Experiments	33
5.1	Experimental Setup	33
5.1.1	Datasets	34
5.1.2	Classifier	34
5.1.3	Implementation Details	35
5.1.4	Evaluation Protocol	35
5.2	Main Results	36
5.3	Ablation Studies	37
5.3.1	Effect of Labeled Ratio	38
5.3.2	Effect of Subgraph Budget	40
5.3.3	Effect of Burning Probability	41
5.3.4	Effect of Subgraph Construction Strategy	42
5.4	Coverage Analysis	44
5.5	Complexity Analysis	47
5.6	Discussion	48
6	Conclusion	51
6.1	Summary of Contributions	51
6.2	Main Findings	52
6.3	Limitations	53
6.4	Future Directions	54
6.5	Closing Remarks	55
	Bibliography	56

List of Tables

2.1	Comparison of common GNN architectures.	10
3.1	Taxonomical overview of GAD with very few labels	25
3.2	Statistics of benchmark dataset in GAD with very few labels	26
4.1	Summary of subgraph construction strategies.	32
5.1	Benchmark dataset statistics	34
5.2	Main results of SFFS augmentation	36
5.3	Ablation ratio - Δ AUC-ROC	39
5.4	Ablation ratio - Δ AUC-PR	39
5.5	Ablation budget - Δ AUC-ROC	40
5.6	Ablation budget - Δ AUC-PR	40
5.7	Ablation burning probability - Δ AUC-ROC	41
5.8	Ablation burning probability - Δ AUC-PR	42
5.9	Ablation construction strategy - Δ AUC-ROC	43
5.10	Ablation construction strategy - Δ AUC-PR	43
5.11	Coverage analysis	46
5.12	Complexity analysis	47

List of Figures

2.1	A graph and its adjacency matrix	4
2.2	Message passing mechanism in GNNs	6
2.3	Node anomalies types in attributed graphs	12
2.4	AUC-ROC and AUC-PR comparison	16
4.1	SFFS-based training augmentation framework pipeline	29
5.1	Anomaly enrichment of <code>fe_generic</code> subgraphs	45

Chapter 1

Introduction

Graph-structured data provide a natural representation for many real-world scenarios, including social networks, financial transactions, and communication infrastructures. In these domains, Graph Anomaly Detection (GAD) aims to identify rare, abnormal patterns that deviate from the expected behavior of the system itself. These anomalies often correspond to critical events such as fraud, cyber-attacks, or the spread of misinformation, which makes their detection an important practical problem.

Despite the progress achieved in recent years, most GAD methods rely on the availability of a sufficiently large set of labeled data. In the real-world, however, labeled anomalies are extremely scarce: they are rare by nature and expensive to annotate, often requiring expert knowledge. This has led to growing interest in weakly supervised settings, where only a very small number of labeled anomalies is available to guide the learning process. A particularly extreme case of this regime is Few-Shot Graph Anomaly Detection (FS-GAD), where models must operate with only a handful of labeled examples - in some cases, as few as one or two.

In this few-label setting, recent methods have explored several learning paradigms, including contrastive learning, meta-learning, prompt tuning, and data-centric strategies. Among these, data-centric strategies have shown promising results by augmenting the training signal through structural patterns, such as metapath-based subgraphs. As an example, "Label-based Graph Augmentation with Metapath for Graph Anomaly Detection" [1] introduces a mechanism to construct local subgraphs around labeled anomalies and use them to enrich node representations through a specialized dual encoder. However, since the subgraph construction and the detection model are jointly trained, it is difficult to assess whether the improvement comes from the structural augmentation itself or from the architecture. This raises a natural question: Can structural subgraph augmentation alone - applied at the data level, independently of the detection model - improve anomaly detection when only a few labels are available?

In this thesis, we address this question by proposing a training augmentation framework based on Subgraph Forest Fire Sampling (SFFS). Given a small set of labeled anomaly seeds, SFFS builds local subgraphs by expanding from each seed to its surrounding neighborhood. The nodes discovered in these subgraphs are added to the training set, enriching the structural context available to the detector without modifying node features or introducing additional architectural components. A GraphSAGE classifier is then trained on the augmented training set.

Our framework is evaluated on six benchmark datasets - Cora, Citeseer, BlogCatalog, Flickr, ACM, and Pubmed - with labeled ratios ranging from 1% to 10% of known anomalies. We also investigate multiple subgraph construction strategies, from free stochastic expansion to schema-guided variants based on metapath patterns, and study their impact through systematic ablation experiments.

Results show that SFFS augmentation consistently improves detection performance across datasets. The gains are particularly evident in AUC-PR, with improvements up to +0.183 on Pubmed and +0.157 on Cora at 10% labeled ratio, while AUC-ROC improvements are more moderate and dataset-dependent.

1.1 Objectives

This thesis pursues the following objectives:

1. **Design a data augmentation framework.** We propose an approach based on Subgraph Forest Fire Sampling (SFFS) that constructs local subgraphs around labeled anomaly seeds and uses them to augment the training set. The goal is to enrich the structural context available during training while keeping the detector architecture unchanged.
2. **Systematic experimental evaluation.** Evaluation of the proposed framework on six benchmark datasets - Cora, Citeseer, BlogCatalog, Flickr, ACM, and Pubmed - using GraphSAGE as the base classifier, with ablation studies covering:
 - labeled anomaly ratios from 1% to 10%;
 - subgraph budgets from 4 to 50 nodes;
 - burning probability from 0.2 to 0.8;
 - subgraph construction strategies.
3. **Analysis of subgraph construction strategies.** We investigate four different strategies for building anomaly-centered subgraphs - `fe_generic`, `fe_path`, `fe_schema`, and `fe_wind` - ranging from unconstrained stochastic

expansion to more structured variants, and study their robustness across different label ratios and graph densities.

The goal is to understand how structural constraints and expansion policies affect the quality of the extracted neighborhoods and the final detection performance. We investigate the robustness of each strategy across different label ratios and graph densities.

1.2 Thesis Structure

The thesis is organized as follows:

- **Chapter 2** introduces the necessary background on graphs, Graph Neural Networks, and graph anomaly detection, including the evaluation metrics used throughout this work.
- **Chapter 3** reviews the state of the art in weakly supervised graph anomaly detection, with a particular focus on few-shot settings. It presents a structured taxonomy of representative methods organized by learning paradigm, together with a discussion of experimental practices in the literature.
- **Chapter 4** presents the proposed framework. It describes the SFFS-based subgraph construction mechanism and the training set augmentation procedure, together with the four subgraph construction strategies investigated in this thesis.
- **Chapter 5** reports the experimental evaluation, organized as follows:
 - Main results on all six benchmark datasets, comparing baseline GraphSAGE with and without SFFS augmentation.
 - Ablation study on the labeled anomaly ratio, ranging from 1% to 10%.
 - Ablation study on the subgraph budget, evaluating budgets from 4 to 50 nodes.
 - Ablation study on the burning probability, evaluating values from 0.2 to 0.8.
 - Ablation study on subgraph construction strategies comparing free stochastic expansion, path-only, schema-guided, and wind-expanded variants.
 - A coverage analysis characterizing the structural properties of SFFS subgraphs.
 - A discussion of the key findings across all experiments.
- **Chapter 6** concludes the thesis with a summary of contributions, an analysis of the limitations of the proposed framework, and directions for future work.

Chapter 2

Background

This chapter introduces the fundamental concepts required to understand the proposed method. We first present graph representations and attributed graphs, which define the basis data setting of this work. We then introduce Graph Neural Networks (GNNs), which are commonly used for learning representations on graph-structured data. Finally, we formalize the GAD problem and describe the few-shot setting, together with the evaluation metrics used in this thesis.

2.1 Graph Fundamentals

A **graph** is a mathematical structure used to model pairwise relationships between objects. Formally, a graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is a set of n nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges representing connections between nodes.

The structure of a graph is typically encoded in its **adjacency matrix** $\mathbf{A} \in \{0,1\}^{n \times n}$, where $A_{ij} = 1$ if there exists an edge between nodes v_i and v_j , and $A_{ij} = 0$ otherwise. For undirected graphs, $\mathbf{A}$ is symmetric.

Figure 2.1 illustrates a simple example of a graph with its corresponding adjacency matrix and node feature matrix.

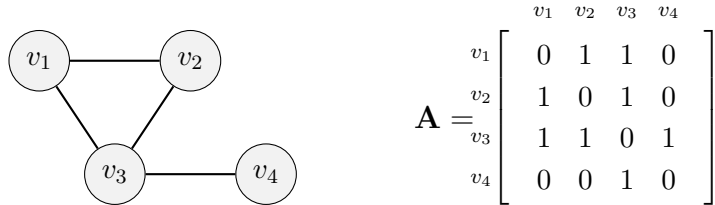


Figure 2.1: Example of an undirected graph (left) and its adjacency matrix $\mathbf{A}$ (right).

Graphs can be categorized according to different properties:

- **Directed and undirected graphs:** in directed graphs, edges have an orientation, while in undirected graphs connections are symmetric.
- **Homogeneous and heterogeneous graphs:** homogeneous graphs contain a single type of nodes and edges, whereas heterogeneous graphs involve multiple node or edge types.
- **Static and dynamic graphs:** static graphs remain fixed over time, while dynamic graphs evolve as nodes, edges, or node attributes change.

These graph categories are commonly adopted in the graph learning literature [2].

In many real-world applications, nodes are not only connected by edges but are also associated with additional descriptive information. This leads to the notion of an **attributed graph**, where each node is represented by a feature vector. Formally, this can be encoded in a feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where d denotes the dimensionality of the node features and each row $\mathbf{x}_i$ corresponds to the attribute representation of node v_i .

Citation networks such as Cora and Citeseer are representative examples of attributed graphs. In these datasets, nodes correspond to academic publications, edges represent citation links, and node features are typically derived from bag-of-words representations of the paper text.

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) are a family of deep learning models designed to operate directly on graph-structured data. Unlike standard neural networks, which assume independent and identically distributed inputs, GNNs explicitly exploit the relational structure of a graph by propagating information between connected nodes. This makes them particularly well suited for tasks where the neighborhood of a node carries discriminative information.

2.2.1 The Message Passing Framework

Most Graph Neural Networks can be described using the message passing framework (Figure 2.2) [3]. The main idea is that, at each layer, a node updates its representation by combining information received from its neighboring nodes.

Given a node v , let $\mathcal{N}(v)$ denote its set of neighbors. At layer l , a message is first computed by aggregating the representations of neighboring nodes:

$$m_v^{(l)} = \text{AGGREGATE} \left(\{h_u^{(l-1)} : u \in \mathcal{N}(v)\} \right) \quad (2.1)$$

The node representation is then updated using both the previous node embedding and the aggregated message:

$$h_v^{(l)} = \text{UPDATE} \left(h_v^{(l-1)}, m_v^{(l)} \right) \quad (2.2)$$

where $h_v^{(l)} \in \mathbb{R}^d$ denotes the embedding of node v at layer l . At the input layer, each node is represented by its original feature vector, i.e., $h_v^{(0)} = x_v$.

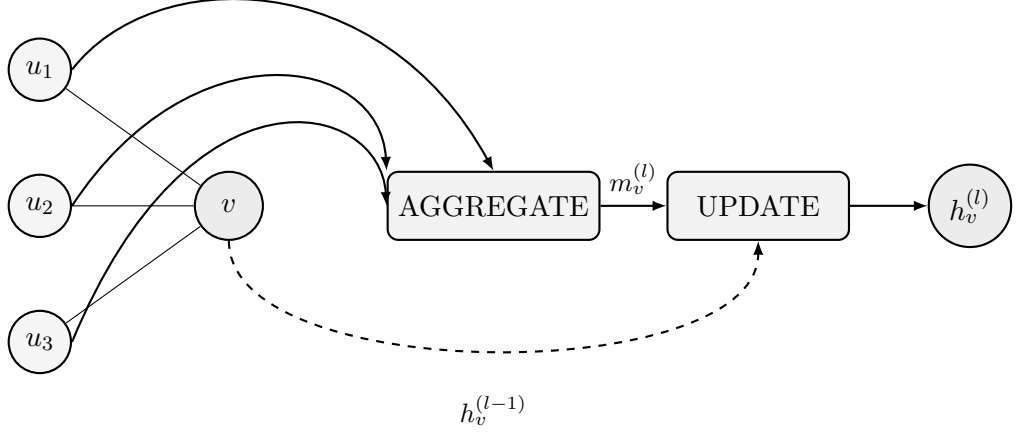


Figure 2.2: Message passing mechanism in GNNs. Neighboring nodes contribute to the aggregation step, while the previous representation of node v is combined in the update function to produce $h_v^{(l)}$.

The choice of the AGGREGATE and UPDATE functions distinguishes different GNN architectures, as we discuss in Sections 2.2.3 and 2.2.4. Common choices for aggregation include sum, mean, and max pooling, while the update step typically involves a learnable linear transformation followed by a non-linear activation.

By stacking multiple layers, each node progressively incorporates information from nodes that are farther away in the graph. After L layers, the representation of a node aggregates information from its entire L -hop neighborhood.

This message passing mechanism allows GNNs to jointly capture structural and attribute information from neighboring nodes. However, repeatedly aggregating neighborhood information across multiple layers may also introduce important limitations. In particular, node representations can become indistinguishable as depth increases, a phenomenon known as *over-smoothing*.

2.2.2 Over-smoothing and its Implications

One of the main limitations of GNNs based on neighborhood aggregation is the phenomenon known as *over-smoothing* [4]. As the number of layers increases, node

representations progressively become more similar to one another, losing their discriminative power.

Intuitively, each GNN layer aggregates information from neighboring nodes. After several layers, a node receives information from a larger portions of the graph, erasing local differences and making nodes difficult to distinguish.

This issue is particularly problematic in GAD. The goal of a GAD model is precisely to distinguish anomalous nodes from their normal neighbors, which requires preserving the local feature differences that make anomalies recognizable. However, if node representations become overly smooth, the differences between anomalous and normal nodes may be suppressed, making anomalies harder to detect.

For this reason, many GAD methods use shallow architectures to limit the over-smoothing effect [5][6].

2.2.3 Graph Convolutional Networks

Graph Convolutional Networks (GCN), introduced by Kipf and Welling [kipf2017semi], are a simple and effective class of GNNs widely used in graph representation learning and form the backbone of the anomaly detectors used in this thesis. They implement a direct form of the message passing framework by iteratively aggregating information from neighboring nodes.

Given an attributed graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, where $\mathbf{A} \in \{0,1\}^{n \times n}$ is the adjacency matrix encoding the graph structure and $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the node feature matrix, self-loops are first added to the adjacency matrix, obtaining $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$.

The degree matrix $\tilde{\mathbf{D}}$ is a diagonal matrix where each entry $\tilde{D}_{ii}$ represents the degree of node v_i in the graph with self-loops.

A single GCN layer is defined as:

$$\mathbf{H}^{(l)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l-1)} \mathbf{W}^{(l)}) \quad (2.3)$$

where $\mathbf{H}^{(l)}$ denotes the node representations at layer l , $\mathbf{W}^{(l)}$ is a learnable weight matrix, and σ is a non-linear activation function.

Intuitively, each layer updates a node representation by combining its own features with those of its neighbors. This operation can be seen as a normalized aggregation of neighboring features, similar to a weighted average where contributions are scaled according to node degrees.

By stacking multiple layers, each node gradually incorporates information from its multi-hop neighborhood. However, as discussed in Section 2.2.2, increasing the depth of the network may lead to over-smoothing, preferring GCN-based models for anomaly detection

2.2.4 Other GNN Variants

While GCN is simple and effective, it also presents some important limitations. First, neighboring nodes are aggregated using coefficients determined by the graph structure and node degrees, rather than learnable weights per edge. Second, each layer aggregates information from the full neighborhood of every node, which can limit scalability on large graphs. These limitations motivated the development of alternative GNN architectures that aim to improve either scalability, flexibility in neighborhood aggregation, or expressive power. In this section, we briefly review three representative models: *GraphSAGE*, *Graph Attention Networks (GAT)*, and *Graph Isomorphism Networks (GIN)*.

GraphSAGE. GraphSAGE [7] was proposed to address the scalability limitations of GCNs on large graphs. In standard GCNs, each node aggregates information from all of its neighbors, which can become computationally expensive for highly connected graphs.

To reduce this cost, GraphSAGE samples only a fixed-size subset of neighbors at each layer. This makes the computation independent of the node degree and enables efficient training on large-scale graphs.

Given a node v , GraphSAGE computes its representation by aggregating information from the sampled neighborhood:

$$\mathbf{h}_v^{(l)} = \sigma\left(\mathbf{W}^{(l)} \cdot \text{AGG}\left(\mathbf{h}_v^{(l-1)}, \{\mathbf{h}_u^{(l-1)} : u \in \mathcal{S}_{\mathcal{N}}(v)\}\right)\right) \quad (2.4)$$

where $\mathcal{S}_{\mathcal{N}}(v)$ denotes the sampled set of neighbors of node v , AGG is a permutation-invariant aggregation function such as mean, sum, or max pooling, and $\mathbf{W}^{(l)}$ is a learnable weight matrix applied at layer l .

Intuitively, instead of using the entire neighborhood, GraphSAGE learns node representations by combining information from a small sampled portion of the graph, followed by a learnable aggregation function. This sampling strategy significantly improves scalability while still preserving local structural information.

Another important advantage of GraphSAGE is its inductive capability, meaning that the model can generalize to unseen nodes or entirely new graphs without requiring retraining.

Graph Attention Networks (GAT). Graph Attention Networks [8] were introduced to overcome one of the main limitations of GCNs: all neighboring nodes contribute to the aggregation with fixed weights determined only by the graph structure.

In many real-world graphs, however, not all neighbors are equally informative. GAT addresses this limitation by introducing a learnable attention mechanism that automatically assigns different importance scores to different neighbors.

For each edge (v, u) , the model computes an attention coefficient α_{vu} that measures how relevant node u is for updating node v :

$$\alpha_{vu} = \text{softmax}_u \left(\text{LeakyReLU} \left(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_v \| \mathbf{W}\mathbf{h}_u] \right) \right) \quad (2.5)$$

where $\mathbf{h}_v$ and $\mathbf{h}_u$ denote the feature representations of nodes v and u , $\mathbf{W}$ is a learnable linear transformation, $\mathbf{a}$ is a learnable attention vector, and $\|$ denotes vector concatenation.

The LeakyReLU activation introduces non-linearity in the attention computation, while the softmax function normalizes the attention coefficients across the neighborhood so that their values sum to one.

The node representation is then updated through a weighted aggregation of neighboring features:

$$\mathbf{h}_v^{(l)} = \sigma \left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu} \mathbf{W}\mathbf{h}_u^{(l-1)} \right) \quad (2.6)$$

Intuitively, GAT allows the model to focus more strongly on the most relevant neighbors while reducing the influence of less informative ones.

Although this attention mechanism improves flexibility and adaptivity, computing attention coefficients for every edge introduces additional computational cost, making GAT less efficient on very large graphs.

Graph Isomorphism Network (GIN). Unlike GraphSAGE and GAT, which mainly focus on scalability and adaptive neighbor weighting, respectively, Graph Isomorphism Networks (GIN) [9] were designed to maximize the expressive power of the message passing framework.

The main motivation behind GIN is that averaging-based aggregation functions, such as those used in the other GNNs, may fail to distinguish different graph structures. To address this limitation, GIN uses sum aggregation combined with a multi-layer perceptron (MLP¹), which provides stronger discriminative capability.

The GIN update rule is defined as:

$$\mathbf{h}_v^{(l)} = \text{MLP}^{(l)} \left((1 + \epsilon) \mathbf{h}_v^{(l-1)} + \sum_{u \in \mathcal{N}(v)} \mathbf{h}_u^{(l-1)} \right) \quad (2.7)$$

where ϵ is a learnable parameter controlling the relative contribution of the central node representation.

¹A Multi-Layer Perceptron (MLP) is a neural network composed of stacked fully connected layers with non-linear activations.

Intuitively, the sum aggregation preserves more structural information than mean-based aggregation, while the MLP enables the model to learn more expressive transformations of the aggregated neighborhood features.

However, the increased expressive power also leads to higher model complexity and a greater risk of overfitting, especially on small datasets.

Summary of GNN Variants. The three architectures discussed above address different limitations of the basic GCN model from complementary perspectives. GraphSAGE improves scalability by restricting neighborhood aggregation through sampling. GAT introduces a learnable attention mechanism that allows the model to assign different importance to different neighbors. GIN maximizes expressive power by combining sum aggregation with MLP-based updates.

A high-level comparison of these architectures is reported in Table 2.1, which summarizes their differences in terms of aggregation strategy, parameterization², scalability³, and expressive power⁴.

Despite these improvements, all the methods discussed above are still based on the message passing paradigm and therefore inherit its fundamental limitations. This shared limitation motivates the design choices of this thesis, where shallow architectures and feature augmentation strategies are adopted to preserve discriminative node-level information.

Table 2.1: Comparison of common GNN architectures.

Model	Aggregation	Weights	Scalability	Expres.
GCN	Mean (normalized)	Fixed	Low	Low
GraphSAGE	Mean/Sum/Max	Fixed	High	Medium
GAT	Attention-weighted mean	Learned	Medium	Medium
GIN	Sum	Fixed	Medium	High

Note: Scalability and expressiveness are qualitative comparisons intended to provide general intuition. Their practical behavior may vary depending on the dataset, graph size, implementation details, and computational resources.

²Parameterization refers to whether the aggregation mechanism depends on learnable parameters (e.g., attention weights) or is fixed and determined only by the graph structure.

³Ability of a model to efficiently handle large graphs in terms of memory and computational cost.

⁴Ability of a model to distinguish nodes or subgraphs with similar local neighborhoods but different connectivity patterns.

2.3 Graph Anomaly Detection

Graph anomaly detection is the task of identifying nodes, edges, or subgraphs that deviate significantly from the expected behavior of the rest of the graph [10, 11, 12]. In many real-world applications - including fraud detection, network intrusion detection, and misinformation identification - anomalies are rare events that, if missed, can have serious consequences.

Most existing GAD methods operate in an *unsupervised* setting, where no labeled examples are available and anomalies are identified purely based on how much a node deviates from the normal patterns of the graph. While effective in many scenarios, unsupervised approaches cannot incorporate prior knowledge about the specific type of anomalies one is looking for. This limitation has motivated growing interest in settings where a very small number of labeled anomalies is available to guide the detection process. These approaches are variously referred to as *weakly supervised* GAD or *few-shot* GAD, depending on the number of labeled examples available and how they are used — this is precisely the setting addressed in this thesis.

2.3.1 Anomalies in Graphs

Compared to traditional anomaly detection, graph-based settings are inherently more complex since anomalies may depend not only on node attributes but also on the relational structure between nodes.

Different types of anomalies can arise in graph data:

- **Node anomalies:** nodes whose structural connectivity or attribute information significantly differs from the surrounding neighborhood. In attributed graphs, this is the most common anomaly detection setting.
- **Edge anomalies:** abnormal or unexpected connections between nodes that do not conform to the overall graph structure.
- **Subgraph anomalies:** groups of nodes and edges that collectively exhibit unusual structural patterns. In this case, the anomalous behavior emerges from the interaction between multiple graph components rather than from a single node.
- **Graph-level anomalies:** entire graphs that significantly deviate from the normal graph distribution. This setting is common in applications where each sample is itself represented as a graph.

In attributed graphs, as explained in [13], anomalies are commonly divided into two main categories, as illustrated in Figure 2.3:

- **Structural anomalies:** nodes, edges, or subgraphs whose connectivity patterns significantly differ from the normal graph structure. Examples include unusually high-degree nodes, unexpected cross-community links, or irregular dense subgraphs.
- **Attribute anomalies:** nodes whose feature vectors are inconsistent with their surrounding context, even if their structural connections appear normal. In attributed graphs, these anomalies are particularly important because they capture discrepancies between node attributes and graph topology.

Since this thesis focuses on attributed graphs and node-level anomaly detection, particular attention is given to anomalies involving node features and their relationship with the local graph topology.

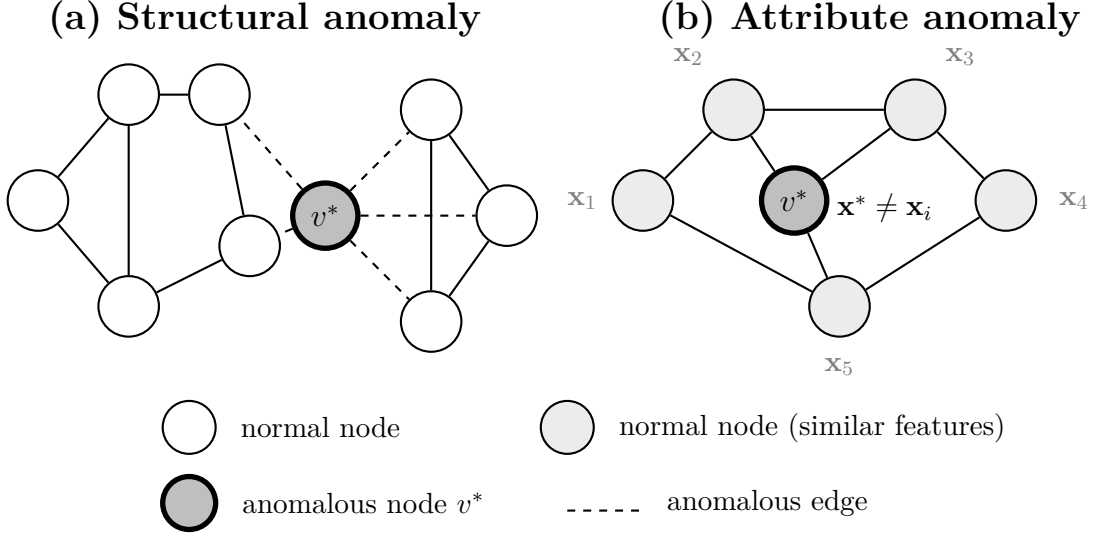


Figure 2.3: Two types of node anomalies in attributed graphs. (a) Structural anomaly: node v^* forms unexpected connections across two separate communities. (b) Attribute anomaly: node v^* has features inconsistent with its neighbors.

2.3.2 Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an attributed graph, where $\mathcal{V}$ is the set of n nodes, $\mathcal{E}$ is the set of edges encoded in the adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the node feature matrix.

Node-level graph anomaly detection aims to assign an anomaly score $s_v \in \mathbb{R}$ to each node $v \in \mathcal{V}$, such that anomalous nodes receive higher scores than normal ones. Formally, the task is to learn a scoring function:

$$f : \mathcal{V} \times \mathcal{G} \rightarrow \mathbb{R} \quad (2.8)$$

such that $f(v, \mathcal{G}) > f(u, \mathcal{G})$ whenever v is anomalous and u is normal. Nodes are then ranked by their score and the top- k are reported as anomalies.

In the weakly supervised setting addressed in this thesis, a small labeled set $\mathcal{L} \subset \mathcal{V}$ of known anomalies is additionally provided, with $|\mathcal{L}| \ll |\mathcal{V}|$. The scoring function f can then exploit this prior knowledge to improve detection, while the majority of nodes remain unlabeled.

2.3.3 Detection Approaches

Graph anomaly detection methods can be broadly categorized according to the level of supervision available during training and the detection principle they rely on.

Unsupervised methods. The majority of GAD methods operate in a fully unsupervised setting, where no labeled anomalies are available.

Among these, *reconstruction-based* approaches are the most widely adopted: a model is trained to reconstruct the graph structure, node attributes, or both, and nodes with high reconstruction errors are flagged as anomalous [14]. Graph autoencoders (GAEs) [15] [16] provide a natural framework for this paradigm, using a GNN encoder to produce node embeddings and a decoder to reconstruct the adjacency matrix or the feature matrix.

Other unsupervised approaches flag nodes that are isolated in the embedding space or structurally distant from their neighbors, based on the intuition that anomalies tend to appear in sparse, poorly connected regions of the graph.

Due to their independence from labeled data, unsupervised methods often lack the prior knowledge needed to distinguish true anomalies of interest from irrelevant outliers or noise, which may lead to a higher number of false positives.

Supervised and semi-supervised methods. When labeled examples are available for both normal and anomalous nodes, graph anomaly detection can be formulated as a standard node classification problem, allowing the model to directly learn discriminative decision boundaries. However, labels are rarely available in practice, especially for anomalies, due to the high cost of manual annotation and the rarity of anomalous events.

Semi-supervised approaches address this limitation by using a small number of labeled nodes together with many unlabeled ones, propagating the label information across the graph to improve detection performance. Although they use fewer labels than fully supervised methods, these approaches still rely on a reasonable amount of labeled anomalies.

Weakly supervised and few-shot methods. More recently, increasing attention has been devoted to settings where only a very limited number of labeled anomalies is available - too few for supervised training but enough to provide useful prior knowledge. Common strategies include contrastive learning, meta-learning, prompt tuning, and data augmentation, often combined to maximize the utility of the scarce labeled information.

Depending on the amount of supervision and the learning strategy adopted, these methods are commonly referred to as *weakly supervised* or *few-shot* graph anomaly detection methods. Since this thesis focuses on this setting, a more detailed discussion of existing approaches is provided in Chapter 3.

2.3.4 Evaluation Metrics

Evaluating graph anomaly detection methods requires metrics that refers to the highly imbalanced nature of the task: anomalous nodes typically represent only a small fraction of the graph, often between 3% and 6% in standard benchmarks, as shown in Table 3.2. In this setting, accuracy is not an informative metric - a classifier that labels every node as normal would still achieve over 94% accuracy on a dataset with 6% anomalies.

Moreover, graph anomaly detectors typically produce continuous anomaly scores rather than binary predictions. For this reason, threshold-independent ranking metrics are generally preferred. This thesis therefore adopts two standard evaluation metrics widely used in anomaly detection: the Area Under the ROC Curve (AUC-ROC) and the Area Under the Precision-Recall Curve (AUC-PR).

AUC-ROC

Graph anomaly detectors typically assign an anomaly score $s_i \in \mathbb{R}$ to each node v_i , where higher values indicate stronger evidence of anomalous behavior. Based on these scores, nodes can be classified as anomalous or normal by applying a decision threshold τ , such that nodes with $s_i \geq \tau$ are predicted as anomalous, while the remaining nodes are considered normal. Rather than evaluating performance at a single threshold, AUC-ROC measures the quality of the ranking induced by the anomaly scores across all possible thresholds.

The Receiver Operating Characteristic (ROC) curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR):

$$\text{TPR}(\tau) = \frac{\text{TP}(\tau)}{\text{TP}(\tau) + \text{FN}(\tau)}, \quad \text{FPR}(\tau) = \frac{\text{FP}(\tau)}{\text{FP}(\tau) + \text{TN}(\tau)} \quad (2.9)$$

where TP, FP, FN, and TN denote true positives, false positives, false negatives, and true negatives respectively. The AUC-ROC summarizes the ROC curve into a

scalar value in $[0,1]$, where 1 corresponds to a perfect ranking and 0.5 to a random classifier. It can also be interpreted as the probability that a randomly selected anomalous node receives a higher anomaly score than a randomly selected normal node [17].

AUC-ROC is widely used because it is threshold-independent and evaluates the relative ordering of anomaly scores rather than their absolute values. However, it has a known limitation in highly imbalanced settings: since normal nodes outnumber anomalies, even a weak detector may achieve a relatively high AUC-ROC by correctly ranking most normal nodes among themselves, while performing poorly on the rare anomalous nodes [18].

AUC-PR

The Precision-Recall (PR) curve is designed to better evaluate performance in highly imbalanced settings by focusing exclusively on the positive class. It plots Precision against Recall across all possible decision thresholds τ :

$$\text{Precision}(\tau) = \frac{\text{TP}(\tau)}{\text{TP}(\tau) + \text{FP}(\tau)}, \quad \text{Recall}(\tau) = \frac{\text{TP}(\tau)}{\text{TP}(\tau) + \text{FN}(\tau)} \quad (2.10)$$

The AUC-PR summarizes this curve into a single scalar in $[0,1]$. Its value depends on the class distribution: in particular, a random classifier achieves an AUC-PR equal to the proportion of positive examples in the dataset (i.e., the anomaly rate π), while a perfect classifier achieves $\text{AUC-PR} = 1$.

Compared to AUC-ROC, AUC-PR is more sensitive to the quality of the top-ranked predictions, since it directly evaluates performance on the positive class. Intuitively, it captures how well the model ranks true anomalies at the top of the ranking while penalizing false positives.

This property makes AUC-PR particularly suitable for highly imbalanced anomaly detection problems, where the main objective is to retrieve the few true anomalies among the top-ranked predictions. For this reason, AUC-PR is widely adopted in the anomaly detection literature, especially in imbalanced graph-based settings [19].

Figure 2.4 illustrates this difference, where two detectors with similar ROC performance show a clear separation in PR space.

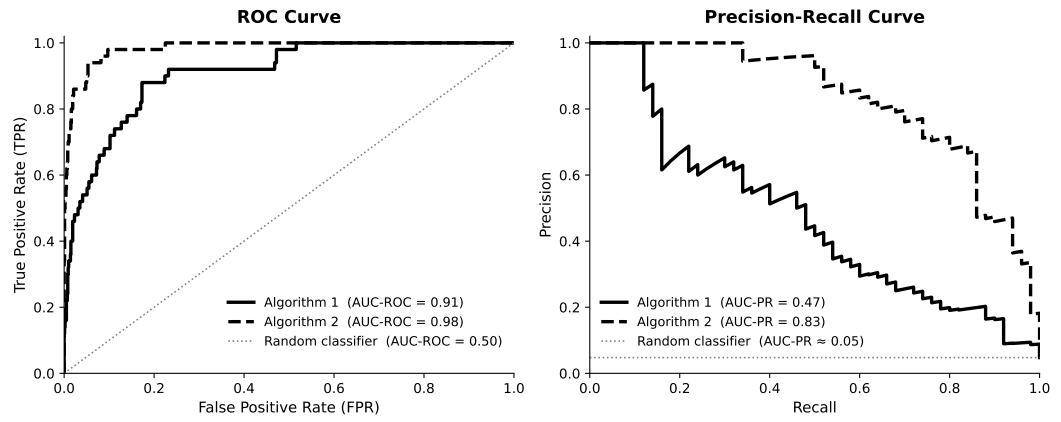


Figure 2.4: Comparison of two anomaly detection algorithms in ROC space (left) and Precision–Recall space (right) on a highly imbalanced dataset with a 5% anomaly rate. Although the two methods appear relatively similar in ROC space, the difference becomes significantly more pronounced in PR space, showing that AUC-PR is more sensitive to the ranking quality of the top-scored anomalous nodes.

Chapter 3

Related Work

This chapter surveys recent methods for graph anomaly detection under limited supervision, with a particular focus on settings where only a very small number of labeled anomalies is available to guide the learning process; a case that in the literature is often referred to as Few-Shot Graph Anomaly Detection. The goal of this chapter is to provide clear overview of how the field deals with this label-scarce setting and to highlight the gap that motivates this thesis. We first describe the methodology used to select and organize the reviewed works, along with the main statistical trends in the field. We then analyze each learning paradigm in detail - contrastive learning, meta-learning, prompt tuning, data-centric learning, and generative, reinforcement, and knowledge distillation approaches - discussing their core ideas, how they adapt to few-label scenarios, and their limitations. The chapter concludes with a discussion of recurring inconsistencies in experimental practices, and places the proposed Feature Enhancement framework within the existing literature.

3.1 Overview of the Literature

This section reports how the relevant literature was collected and summarizes the main trends observed across the surveyed works in terms of methods, graph settings, application domains, and evaluation practices.

3.1.1 Selection Methodology

To provide a comprehensive overview, we conducted a systematic literature search across major academic platforms, including Google Scholar, arXiv, ACM Digital Library, and IEEE Xplore.

The search strategy and the query combined the core terms “graph anomaly

detection” or “GAD” with supervision-related terms such as “few labels”, “limited labels”, “weakly supervised”, and “few-shot”.

Papers were selected based on four criteria: relevance to FS-GAD, methodological novelty, publication recency, and clarity of experimental evaluation.

Following this procedure, we identified a curated set of 20 representative works. Each paper is described in Table 3.1 in terms of method category, publication year, graph type, anomaly granularity, application domain, core idea, and evaluation metrics, allowing for a consistent comparison across heterogeneous approaches.

It is important to note that while Table 3.1 assigns each paper to a single primary category for taxonomic clarity, many methods are inherently hybrid. For instance, meta-learning and prompt tuning works often integrate contrastive learning objectives to refine embedding representations. Our classification prioritizes the core methodological paradigm or primary contribution of the paper, as emphasized by the authors.

3.1.2 Key Trends and Statistics

From the collected works, several key trends emerge.

- **Methodological trends.** Contrastive learning is the dominant paradigm, followed by meta-learning and prompt-based approaches. Data-centric and generative methods appear less frequently, while reinforcement learning and knowledge distillation remain relatively marginal. Overall, most approaches rely on representation learning and adaptation techniques to cope with limited labeled data.
- **Graph settings and anomaly granularity.** Most surveyed methods focus on static graphs (19 out of 20 works), indicating that dynamic settings remains largely underexplored. Node-level detection is also predominant (16 out of 20 methods), whereas edge-level and graph-level anomaly detection receive significantly less attention.
- **Application domains and datasets.** The majority of methods are evaluated on general-purpose benchmark datasets rather than domain-specific scenarios, as also shown in Table 3.2. Only a limited number of works target specific applications such as financial fraud detection, blockchain analysis, or cybersecurity.
- **Evaluation practices.** AUC-ROC and AUC-PR are the most commonly used metrics, mainly because of the strong class imbalance typical of anomaly detection tasks, as discussed in Section 2.3.4.

3.2 Learning techniques

This section describes the main learning approaches adopted in the surveyed literature for graph anomaly detection under very few labels, grouped by their dominant methodological paradigm. As summarized in Table 3.1, existing methods can be broadly categorized into contrastive learning, meta-learning, prompt tuning, data-centric learning, generative models, reinforcement learning, and knowledge distillation. Each paradigm tackles the label-scarce setting from a different perspective, and the following subsections present their core principles and design patterns.

3.2.1 Contrastive Learning

Contrastive learning is the most widely adopted paradigm in the surveyed literature. Its main idea is to learn node representations by bringing similar samples closer in the embedding space while pushing dissimilar ones apart. This makes it particularly suitable for settings where only a few labeled examples are available.

A common design choice across existing methods is the use of multiple augmented views of the same graph. By applying perturbations such as edge dropping, feature masking, or subgraph sampling, models learn to preserve invariant structural and semantic information while being robust to noise. In this context, the supervision from a small number of labeled anomalies is typically used to guide the separation between normal and anomalous representations, rather than to directly train a classifier.

Several works extend this general idea in different directions. FMGAD [22] combines contrastive learning with a reconstruction objective, using multiple views of local subgraphs to enforce consistency and propagate anomaly signals to unlabeled nodes. SADMCL [21] introduces a multi-level contrastive framework that operates at multiple levels — sample, instance, and class — building explicit prototypes for normal and anomalous nodes to improve separability under weak supervision. Similarly, CGNN [23] integrates contrastive learning in a context-aware representation space, where node semantics are decomposed into category-specific subspaces to better separate fraudulent behavior from normal patterns.

Other approaches adapt the contrastive paradigm to more specific tasks. FEAE [24] focuses on edge-level anomalies and combines contrastive separation of real and corrupted edges with a reconstruction-based objective, while AD-GSCL [20] extends the framework to rumor detection by combining supervised contrastive learning on labeled data with self-supervised learning on unlabeled data.

Overall, contrastive methods show high performance in few-label settings. However, their effectiveness strongly depends on the quality of graph augmentations and on how positive and negative pairs are defined. In many cases, the notion of

“similarity” is task-dependent, which can limit their ability to generalize across different domains.

3.2.2 Meta-Learning

The core idea of meta-learning approaches is to learn transferable knowledge across multiple tasks or graphs, so that the model can quickly adapt to a new target graph with only a few labeled anomalies and effectively “learn how to learn” from limited supervision.

Most approaches in this category are built on episodic training strategies inspired by MAML [39]¹, where the model is repeatedly trained on different few-shot tasks in order to develop a general ability to adapt to new GAD scenarios. MA-GAD [25] applies this idea in graph anomaly detection by training a model that can rapidly adapt to new tasks after a few gradient steps, while also using graph compression² to reduce redundancy in the learned representations. MetaGAD [26] combines meta-learning with self-supervised learning using a bi-level optimization framework. This helps the model learn shared representations across different tasks while adapting them to each specific task, improving robustness of the learned embeddings.

Other approaches focus on making the framework more robust to noise and variability in graph data. GMFITD [27] introduces a more flexible modeling of the graph structure, where connections and relationships are treated in a probabilistic way rather than as fixed, which helps the model better handle uncertain or noisy labels. This is particularly relevant in domains such as cybersecurity, where annotations are often unreliable. DARGAT-GDN [28] extends this idea to cross-graph settings, enabling knowledge transfer between different datasets while preserving the ability to distinguish between normal and anomalous nodes.

Overall, meta-learning methods work well in the few-label setting because they are designed for fast adaptation. However, they often require building multiple auxiliary tasks or graphs, which can be expensive and may not always be possible when only one target graph is available.

3.2.3 Prompt Tuning

Prompt tuning methods leverage the power of pre-trained models to avoid fully retraining on scarce target data. The main idea is to keep the pre-trained encoder frozen and introduce a small set of learnable parameters, called prompts, to provide

¹MAML (Model-Agnostic Meta-Learning) is a meta-learning framework that trains a model to find an initialization from which it can adapt to a new task using only a few gradient updates.

²Graph compression refers to reducing redundant or less informative parts of the graph structure to improve efficiency and reduce noise during training.

task-specific information. This allows the model to adapt to different graphs and anomaly types even with few labeled examples, making this approach particularly effective in few-label settings.

AffinityTune [29] reformulates anomaly detection as an affinity judgment problem, introducing prompt tokens that capture node relationships at different levels of granularity and using contrastive learning to improve separation between normal and anomalous patterns. CDFS-GAD [30] extends this idea to cross-domain settings, where prompts are used to adapt the model to new graph distributions while maintaining consistency across domains. AnomalyGFM [32] proposes a graph foundation model that learns general prototypes for normal and anomalous patterns during pre-training, and refines only the normal prototype at inference time using a small set of labeled nodes, leaving the encoder unchanged. AnomalyLLM [31] aligns graph representations with the embedding space of a large language model (LLM), using in-context prompts that include a few labeled examples to guide the classification of new edges without any parameter updates. This makes the model suitable for dynamic or evolving graph scenarios.

Overall, prompt tuning methods are attractive because they require little supervision without fully retraining the base model. However, their performance strongly depends on the quality of the pre-trained model and on how well the prompts capture task-specific patterns, which can limit their effectiveness when applied to very different graph settings.

3.2.4 Data-Centric Learning

Data-centric approaches focus on improving the quality and diversity of the training data rather than designing more complex models. The main idea is to use the available labeled anomalies to construct additional structural or synthetic information that enriches node representations and makes normal and anomalous patterns more separable.

Among the surveyed methods, MGAD [1] is the most directly relevant to this thesis, as discussed in Section 3.4. It assigns node types based on available labels - anomalous or unknown - and generates metapath-based subgraphs via random walks that match specific structural patterns around labeled anomalies. These subgraphs are then processed by a dual encoder that combines global and local structural information. EnGraph [33] combines graph embeddings with a GAN-based augmentation step that generates pseudo-anomalies; then aggregates multiple base detectors through ensemble scoring to produce a robust final anomaly ranking. MSLP [34] follows a different direction by propagating a small number of labeled anomalies in blockchain transaction graphs and filtering the resulting pseudo-labels through a confidence-based selection mechanism to build a cleaner and more balanced training set.

Overall, data-centric methods show that improving or expanding the training data can be as important as model design in few-label settings. However, the quality of the augmented data depends heavily on the availability and reliability of the few labeled anomalies, and poorly designed augmentations can introduce noise that may reduce performance.

3.2.5 Generative, Reinforcement, and Knowledge Distillation

The remaining methods in the survey are less common and explore alternative strategies to address the GAD under very few labels.

Two works leverage generative models to address the scarcity of labeled anomalies by creating additional synthetic training examples. GenGA [35] uses a diffusion model that generates synthetic node attributes based on the graph structure, augmenting the training set of existing detectors to improve their ability to learn a better decision boundary under extreme class imbalance. GUDI [36] pre-trains a diffusion model in a self-supervised way on large unlabeled graphs. At inference time, a lightweight classifier is used to guide the model in identifying anomalous nodes, allowing it to generalize to new domains without additional fine-tuning.

Reinforcement learning approaches treat anomaly detection as a sequential decision problem. REGAD [37] learns a policy that identifies and prunes suspicious edges in the graph, optimizing detection performance through reward signals.

Knowledge distillation methods focus on transferring knowledge from a larger model (teacher) to a smaller and simpler one (student). Dual-KD [38] follows this idea, where a pre-trained teacher model guides a student on the target graph using both its predictions and internal representations, enabling effective learning even with only a few labeled anomalies.

3.3 Problem Settings and Experimental Issues

Despite the growing number of works on few-label GAD, there are still several inconsistencies and ambiguities in experimental settings, which limit a fair comparison across methods. In this section, we summarize the most common issues that appear in the literature.

Labeling assumptions and unlabeled data. A recurring ambiguity concerns the labeling of nodes outside the anomalous set. In many studies, it is unclear whether the remaining nodes are entirely unlabeled or partially annotated as normal. Some works explicitly adopt a Positive-Unlabeled (PU) learning setting [22], where

only anomalous nodes are labeled and all others are treated as unlabeled, while others introduce the concept of gray samples to denote nodes of uncertain class [23].

Few-shot definitions and dataset scale. Another important issue is how few-shot settings are defined. Most works use fixed numbers of labeled anomalies (e.g., 3-shot, 5-shot, 10-shot), while others rely on ratio-based definitions. However, absolute shot settings are not scale-invariant: the same number of labeled anomalies can represent very different supervision levels depending on the graph size. Moreover, the literature rarely defines a clear boundary between a true few-shot setting and a more general limited-label scenario, making comparisons across works less consistent. For this reason, some studies instead use percentage-based ratios, which better reflect the relative amount of supervision and allow more consistent comparisons across datasets [1].

Class imbalance. Class imbalance is a central challenge in few-label graph anomaly detection, as anomalous nodes are typically rarer than normal ones. Some studies preserve the original imbalance during sampling [23], while others explicitly control it through imbalance ratios used to guide a cost-sensitive loss functions that assign higher weight to anomalies [26]. However, maintaining realistic imbalance while ensuring sufficient training examples remains a non-trivial issue.

Noisy labels and contamination. Few-label settings are also highly sensitive to noisy labels and data contamination. In real-world graphs, some labeled anomalies may be incorrect, and unlabeled nodes may still contain hidden anomalies. Even a small amount of noise can significantly degrade performance. To study this, some works explicitly introduce noise ratios [37] or contamination ratios [26] to evaluate model robustness under imperfect supervision.

3.4 Positioning of this Work

The methods presented in this chapter address graph anomaly detection with limited labeled data through a variety of learning strategies. Despite their differences, most of these methods are designed for a specific model or training strategy, making it difficult to evaluate the contribution of individual components.

Among them, the method proposed in [1] is the most closely related to this thesis, as it shares the core intuition that structural proximity to labeled anomalies can provide useful information even when only a few labels are available. Their idea is based on the generation of metapath-based subgraphs around labeled anomalies via random walks and processes them through a dual encoder that combines global graph information with local anomaly-aware context.

The proposed framework shares this intuition but takes a different approach. Rather than designing a new detection architecture, we use SFFS-generated subgraphs to augment the training set of a GraphSAGE classifier. Nodes structurally close to known anomalies are added to the training set, providing additional structural context during learning. This design allows us to evaluate the contribution of structural neighborhood augmentation independently of the detection architecture.

Table 3.1: Taxonomical overview of the surveyed papers in GAD under very limited supervision.

Paradigm	Method	Year	Graph	Anomaly	Domain	Core Idea	Metrics
Contrastive Learning	AD-GSCL [20]	2025	Static	Graph	Rumor	FS-GAD via supervised and unsupervised contrastive learning	Acc, Pre, Rec, F1, AUC
	SADMCL [21]	2024	Static	Node	Generic	Multi-view contrastive learning to separate the two classes	AUC-ROC, Pre, Rec
	FMGAD [22]	2023	Static	Node	Generic	Multi-view contrastive with message-enhanced reconstruction	AUC-ROC, AUC-PR
	CGNN [23]	2025	Static	Node	Financial	Contrastive learning on category-specific subspaces	F1, Pre, Rec
	FEAE [24]	2025	Static	Edge	Net. Security	Hybrid self-supervised contrastive and reconstruction learning	F1, Pre
Meta Learning	MA-GAD [25]	2025	Static	Graph	Biochemical	MAML meta-learning with graph compression	AUC-ROC
	MetaGAD [26]	2024	Static	Node	Generic	Bi-level meta-learning with RAN adapting self-supervised embeddings	AUC-ROC, AUC-PR
	GMFITD [27]	2024	Static	Node	Cybersecurity	Episodic meta-learning with soft edges and adversarial pruning	Acc, F1
	DARGAT-GDN [28]	2025	Static	Node	Generic	Cross-graph meta-learning with node-differentiated attention	AUC-ROC, AUC-PR
Prompt Tuning	AffinityTune [29]	2025	Static	Node	Generic	Prompt-tuning with multi-granularity contrastive learning	AUC-ROC, AUC-PR
	CDFS-GAD [30]	2024	Static	Node	Generic	Cross-domain few-shot with domain-adaptive contrastive, prompt-tuning	AUC-ROC, AUC-PR
	AnomalyLLM [31]	2024	Dynamic	Edge	Generic	In-context prompts after GNN-LLM alignment	AUC
	AnomalyGFM [32]	2025	Static	Node	Generic	Graph foundation model with residual prototypes and prompt-tuning	AUC-ROC, AUC-PR
Data-Centric Learning	EnGraph [33]	2025	Static	Node	Generic	Pseudo-anomaly generation and ensemble of base detectors	AUC-ROC, AUC-PR, F1
	MGAD [1]	2024	Static	Node	Generic	Metapath-guided subgraph augmentation with dual encoder	AUC
	MSLP [34]	2024	Static	Node	Financial	Minimal substitution label propagation	F1, Rec
Generative Learning	GenGA [35]	2024	Static	Node	Generic	Diffusion-based synthetic node generation	F1, AUC, AUC-PR
	GUDI [36]	2024	Static	Node	Generic	Domain-agnostic diffusion pre-training with few-shot adaptation	AUC-ROC, Rec@k
Reinforcement Learning	REGAD [37]	2024	Static	Node	Generic	GCN-based RL policy to prune suspicious edges	AUC, AUC-PR
Knowledge Distillation	Dual-KD [38]	2025	Static	Node	Generic	Dual-level distillation via soft predictions and representation alignment	AUC-ROC, AUC-PR, Pre@K

Note: **AUC-ROC**: Area Under the ROC Curve; **AUC-PR**: Area Under the Precision-Recall Curve; **Acc**: Accuracy; **Pre**: Precision; **Rec**: Recall; **F1**: F1-score.

Table 3.2: Statistics of benchmark datasets used in the surveyed GAD under very limited supervision.

Dataset	Domain	#Nodes	#Edges	#Features	#Anomalies	Ratio	Type	#Papers
<i>Citation Networks</i>								
Cora	Citation	2,708	11,606	1,433	150	5.5%	Inj.	6
Citeseer	Citation	3,327	10,154	3,703	150	4.5%	Inj.	5
Pubmed	Citation	19,717	44,338	500	600	3.0%	Inj.	3
ACM	Citation	16,484	71,980	8,337	600	3.6%	Inj.	3
<i>Social Networks</i>								
BlogCatalog	Social	5,196	171,743	8,189	300	5.8%	Inj.	5
Reddit	Social	10,984	168,016	64	366	3.3%	Org.	5
Flickr	Social	7,575	239,738	12,047	450	5.9%	Inj.	4
Weibo	Social	8,405	407,963	400	868	10.3%	Org.	4
Wiki	Social	9,227	18,257	64	217	2.4%	Org.	3
Tolokers	Social	11,758	519,000	10	2,563	21.8%	Org.	2
Questions	Social	48,921	153,540	301	1,467	3.0%	Org.	2
Facebook	Social	1,081	55,104	576	25	2.3%	Org.	1
<i>E-commerce and Review Networks</i>								
Amazon	E-commerce	13,752	515,042	767	694	5.0%	Inj.	5
YelpChi	Review	45,954	3,846,979	32	6,677	14.5%	Org.	4
Disney	Co-purchase	124	335	28	6	4.8%	Org.	2
<i>Financial and Blockchain Networks</i>								
T-Finance	Financial	39,357	21,222,543	10	1,803	4.6%	Org.	3
Elliptic	Blockchain	46,564	73,248	93	4,545	9.8%	Org.	3
DGraphFin	Financial	3,700,550	—	—	—	1.3%	Org.	1
<i>Other Domains</i>								
T-Social	Social	5,781,065	73,105,508	10	174,280	3.0%	Org.	2
CERT	Cybersecurity	—	—	—	—	—	Org.	1

Note: **Type:** *Inj.* = injected anomalies, *Org.* = organic anomalies. **#Papers** = number of surveyed works using the dataset. **Ratio** = proportion of anomalous nodes over total nodes.

Chapter 4

Proposed Method

This chapter presents the proposed framework for graph anomaly detection with very few labels. The core idea is to exploit a small set of labeled anomaly seeds to construct local subgraphs around them and use these subgraphs to augment the training set. This augmentation is performed through Subgraph Forest Fire Sampling (SFFS) [40], a stochastic expansion mechanism that builds local neighborhoods around each labeled seed and adds the discovered nodes to the training set before model training. The approach requires no modification to the detector architecture and no additional feature engineering.

Building on the GAD formulation introduced in Section 2.3, we assume that a small set of labeled anomaly seeds $\mathcal{S} \subset \mathcal{V}_{\text{train}}$ is available, with $|\mathcal{S}| \ll |\mathcal{V}|$. The goal is to exploit this limited supervision to enrich the training set with nodes that are structurally close to known anomalies, providing the detector with a richer and more informative training context.

4.1 Overview

The proposed pipeline is illustrated in Figure 4.1 and consists of four main stages.

The method starts from an attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ and assumes the availability of a small set of labeled anomaly seeds $\mathcal{S} \subset \mathcal{V}_{\text{train}}$, sampled only from the training partition. The goal is to exploit these few labels to enrich the training data with structurally relevant nodes.

1. **SFFS subgraph construction.** For each labeled seed $v_s \in \mathcal{S}$, a local subgraph $\mathcal{S}_s$ is generated using Subgraph Forest Fire Sampling (SFFS), a stochastic expansion process that explores the neighborhood of the seed (Section 4.2.1). The expansion is controlled by a burning probability p and a budget B .

2. **Training set augmentation.** The nodes collected in the subgraphs are added to the original training set, producing an augmented training set

$$\mathcal{V}'_{\text{train}} = \mathcal{V}_{\text{train}} \cup \bigcup_{k=1}^{|\mathcal{S}|} \mathcal{S}_k \quad (4.1)$$

Node features remain unchanged, and only the set of training nodes is modified.

3. **GraphSAGE training.** A GraphSAGE-based binary classifier [7] is trained on the augmented training set. The model performs message passing over the graph structure and learns to distinguish anomalous from normal nodes using the available labels.
4. **Anomaly scoring.** After training, the model assigns an anomaly score s_i to each node $v_i \in \mathcal{V}_{\text{test}}$. Evaluation is performed exclusively on the original test partition, which remains untouched during training.

This design ensures that the SFFS procedure acts purely as a *data-level augmentation mechanism*, without modifying the GraphSAGE architecture or introducing additional feature engineering. This allows us to isolate and evaluate the contribution of structural neighborhood augmentation on detection performance independently of model design choices.

4.2 Subgraph Construction via SFFS

Given the set of labeled anomaly seeds $\mathcal{S}$, the first step of the pipeline is to construct a local subgraph around each seed node. These subgraphs define the structural neighborhoods whose nodes will be added to the training set.

4.2.1 Forest Fire Sampling Mechanism

We adopt a stochastic expansion procedure inspired by the Forest Fire graph generation model [40]. Starting from a seed node $v_s \in \mathcal{S}$, a subgraph is constructed by recursively expanding to unvisited neighbors in a probabilistic manner.

At each step, a fraction of neighbors is selected based on a stochastic burning process. Specifically, a value ρ is drawn from a triangular distribution:

$$\rho \sim \text{Triangular}(0, p, 1)$$

where $p \in (0,1)$ is the burning probability that controls the expansion intensity. The number of neighbors selected for expansion is then:

$$x = \max(1, \min(\lfloor \rho \cdot |\mathcal{N}_u(v)| \rfloor, B - |\mathcal{S}_s|)) \quad (4.2)$$

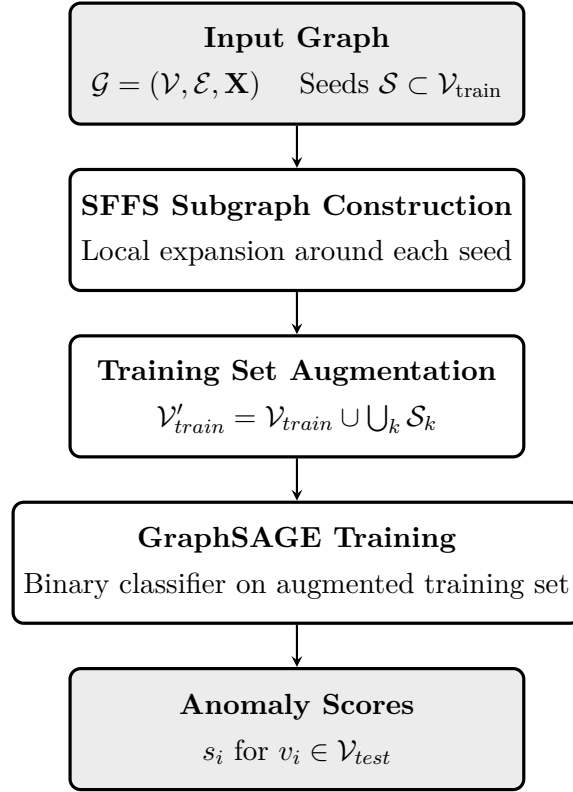


Figure 4.1: Overview of the proposed SFFS-based training augmentation framework. Starting from an input graph, a small set of labeled anomaly seeds is selected from the training partition. Local subgraphs are constructed via SFFS and used to augment the training set. A GraphSAGE model is then trained on the augmented graph and evaluated on the test set.

where $\mathcal{N}_u(v)$ denotes the unvisited neighbors of node v and B is the subgraph budget.

The process iterates until either the budget B is reached or no further expansion is possible. The parameter p controls how the subgraph grows: small values of p produce thin, path-like subgraphs, while large values lead to wider and more connected subgraphs.

Compared to the original Forest Fire model [40], which generates entire graphs through recursive stochastic expansion, our procedure samples a local subgraph from an existing graph starting from a fixed anomaly seed. Two key design choices are particularly important in our setting:

- **Per-seed locality.** One independent subgraph is constructed per labeled seed, each with its own budget B , rather than a single global structure shared across

all seeds. This preserves the structural locality of the sampled neighborhood around its anomaly seed.

- **Connected termination.** Expansion stops when no unvisited neighbors remain, rather than restarting from a random node to fill a global budget. This ensures that each subgraph remains structurally connected to its original seed node.

Algorithm 1 summarizes the construction procedure.

Algorithm 1 SFFS Subgraph Construction. Starting from a seed node, the fire spreads stochastically through the graph until the budget B is reached or the queue is exhausted. The burning probability p controls the breadth of the expansion.

```

1: procedure SFFS( $\mathcal{G}, v_s, p, B$ )
2:    $\triangleright \mathcal{G} = (\mathcal{V}, \mathcal{E})$  is the input graph
3:    $\triangleright v_s \in \mathcal{S}$  is the labeled anomaly seed
4:    $\triangleright p \in (0,1)$  is the burning probability
5:    $\triangleright B$  is the subgraph budget
6:    $\triangleright$  Initialization
7:    $\mathcal{S}_s \leftarrow \{v_s\}$   $\triangleright$  subgraph initialized with seed
8:   queue  $\leftarrow [v_s]$ 
9:   visited  $\leftarrow \{v_s\}$ 
10:   $\triangleright$  Expansion
11:  while queue  $\neq \emptyset$  and  $|\mathcal{S}_s| < B$  do
12:     $v \leftarrow \text{queue.popleft}()$ 
13:     $\mathcal{N}_u(v) \leftarrow \{u \in \mathcal{N}(v) \mid u \notin \text{visited}\}$   $\triangleright$  unvisited neighbors
14:    if  $\mathcal{N}_u(v) = \emptyset$  then
15:      continue
16:    end if
17:    pct  $\sim \text{Triangular}(0, p, 1)$   $\triangleright$  sample burning fraction
18:     $x \leftarrow \max(1, \min(\lfloor \text{pct} \cdot |\mathcal{N}_u(v)| \rfloor, B - |\mathcal{S}_s|))$   $\triangleright$  neighbors to burn
19:     $V_x \leftarrow \text{sample}(\mathcal{N}_u(v), x)$   $\triangleright$  randomly select  $x$  neighbors
20:    for  $u \in V_x$  do
21:       $\mathcal{S}_s \leftarrow \mathcal{S}_s \cup \{u\}$ 
22:      visited  $\leftarrow \text{visited} \cup \{u\}$ 
23:      queue.append( $u$ )
24:    end for
25:  end while
26:  return  $\mathcal{S}_s$   $\triangleright$  local subgraph centered around  $v_s$ 
27: end procedure

```

4.2.2 Subgraph Construction Strategies

We investigate four strategies for constructing subgraphs around labeled seeds, ranging from unconstrained stochastic expansion to schema-constrained variants. All methods build on the SFFS procedure described in Section 4.2.1 and differ in how structural constraints are enforced during expansion. Their main properties are summarized in Table 4.1.

fe_generic

The **fe_generic** strategy applies the SFFS procedure without any structural constraint on node types. Starting from a seed node, the expansion proceeds freely until the budget B is reached or no further neighbors are available. This strategy is fully robust and always produces a valid subgraph for each seed. On sparse graphs, the final subgraph may contain fewer than B nodes due to limited neighborhood availability.

fe_path

The **fe_path** strategy constructs subgraphs by searching for predefined schema paths that describe specific sequences of anomalous (A) and normal (N) nodes.

We consider three schemas:

- ANA: anomaly $\rightarrow$ normal $\rightarrow$ anomaly;
- ANNA: anomaly $\rightarrow$ normal $\rightarrow$ normal $\rightarrow$ anomaly;
- ANAN: anomaly $\rightarrow$ normal $\rightarrow$ anomaly $\rightarrow$ normal.

Starting from a seed node, the search explores the graph step by step, selecting at each stage a neighbor whose label matches the schema requirements. If a complete valid path is found, the nodes along the path form the resulting subgraph, with no additional expansion. If no valid path exists in the local neighborhood of the seed, the seed is discarded and does not contribute a subgraph. As a consequence, the strategy may fail when the required schema pattern is not present in the local neighborhood of the seed.

fe_schema

The **fe_schema** strategy combines stochastic expansion with schema verification. A subgraph is first generated using SFFS with a depth constraint consistent with the selected schema (ANA, ANNA, or ANAN). The schema constraint is then verified a posteriori: the subgraph is kept only if it contains at least one valid schema path. This strategy has a higher failure risk on sparse graphs, where the expansion may

not encounter a valid schema path within the depth constraint, especially at low labeled ratios.

fe_wind

The **fe_wind** strategy first guarantees the existence of a valid schema path (ANA, ANNA, or ANAN) through repeated random walk attempts (maximum 200). Once a valid path is found, the subgraph is expanded around it using the SFFS procedure until the budget B is reached. This design ensures both schema validity and controlled subgraph size, providing a more robust and reliable construction strategy. It combines the strict structural constraints of **fe_path** with the richer neighborhood coverage of **fe_generic**. The strategy may still fail when no valid schema path can be identified after the maximum number of attempts, although this occurs rarely in practice.

Table 4.1: Summary of subgraph construction strategies.

Strategy	Schema	Budget	Avg. size	Failure risk
fe_generic	None	Fixed B	$\leq B$	None
fe_path	Yes	Path only	3–4 nodes	Medium
fe_schema	Yes	Depth-limited	Variable	High (sparse/dense)
fe_wind	Yes	Fixed B	$\leq B$	Low

Chapter 5

Experiments

This chapter presents the experimental evaluation of the proposed SFFS training augmentation framework. The goal is to assess whether enriching the training set with nodes structurally close to known anomalies can improve graph anomaly detection under limited supervision, and to analyze the factors influencing the effectiveness of the augmentation.

We first describe the experimental setup, including datasets, the GraphSAGE classifier, implementation details, and evaluation protocol. We then report the main results on six benchmark datasets, followed by ablation studies investigating the effect of the main components of the framework: labeled ratio, subgraph budget, burning probability, and construction strategy. Finally, we present a coverage analysis of the generated subgraphs and a complexity analysis of the additional computational cost introduced by the augmentation step.

Unless otherwise specified, experiments use the default configuration: a subgraph budget of $B = 20$, burning probability $p = 0.6$, labeled anomaly ratio $\rho = 10\%$, and the `fe_generic` construction strategy.

Due to the strong class imbalance characterizing the anomaly detection setting, the discussion of the results places particular emphasis on AUC-PR, while AUC-ROC is also reported as a complementary measure of ranking performance.

5.1 Experimental Setup

This section describes the experimental setup used throughout the evaluation of the proposed SFFS training augmentation framework. We first present the benchmark datasets, followed by the classifier, implementation details, and evaluation protocol.

5.1.1 Datasets

We evaluate the proposed framework on six benchmark datasets commonly used in graph anomaly detection: Cora, Citeseer, BlogCatalog, Flickr, ACM, and Pubmed. The datasets include both citation networks and social networks, covering a broad range of graph sizes, feature dimensionalities, and structural properties. All datasets follow the anomaly injection protocol adopted in [5], where anomalies are synthetically injected through two mechanisms:

- **Structural anomalies:** selected nodes are connected together to form dense cliques, creating unusual structural patterns in the graph.
- **Attribute anomalies:** selected nodes have their feature vectors replaced with those of semantically distant nodes, creating abnormal attribute patterns.

The two types are injected independently and do not overlap, meaning that each anomalous node is either a structural or an attribute anomaly, but not both.

Table 5.1 reports the key statistics of the six datasets used in our experiments.

Table 5.1: Statistics of the six benchmark datasets used in our experiments. Anomalies are synthetically injected

Dataset	Nodes	Edges	Features	Anomalies	Ratio
Cora	2,708	5,429	1,433	150	5.5%
Citeseer	3,327	4,732	3,703	150	4.5%
BlogCatalog	5,196	172,759	8,189	298	5.7%
Flickr	7,575	241,277	12,047	445	5.9%
ACM	16,484	82,175	8,337	597	3.6%
Pubmed	19,717	46,424	500	600	3.0%

The six datasets cover a variety of graph types and sizes, including both citation networks (Cora, Citeseer, ACM, and Pubmed) and social networks (BlogCatalog and Flickr). They differ in the number of nodes, edges, feature dimensions, and anomaly ratios, providing a diverse benchmark for evaluating the proposed method. In particular, Cora and Citeseer are relatively small and sparse graphs, while BlogCatalog and Flickr are much denser. ACM and Pubmed represent the largest datasets considered in our experiments.

5.1.2 Classifier

We use GraphSAGE [7] as the base classifier for all experiments. A detailed description of the model is provided in Section 2.2.4.

GraphSAGE was selected because its inductive nature allows a clear separation between training and test nodes, making it particularly suitable for evaluating the effect of the proposed training augmentation strategy.

The architecture consists of two GraphSAGE layers with hidden dimension 64 and output dimension 32, followed by a linear classification layer. ReLU activations are applied after each GraphSAGE layer and a dropout layer with rate 0.3 is inserted after the first layer.

The model is trained as a binary classifier using a weighted Binary Cross-Entropy loss:

$$\mathcal{L} = -\frac{1}{|\mathcal{V}'_{\text{train}}|} \sum_{v_i \in \mathcal{V}'_{\text{train}}} [w \cdot y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \quad (5.1)$$

where $w = n_{\text{neg}}/n_{\text{pos}}$ is the positive class weight to account for class imbalance, $y_i \in \{0,1\}$ is the ground-truth label, and $\hat{y}_i \in [0,1]$ is the predicted anomaly probability.

Given the augmented training set $\mathcal{V}'_{\text{train}}$ produced by SFFS, GraphSAGE is trained using only the nodes contained in $\mathcal{V}'_{\text{train}}$. Both message passing and loss computation are restricted to this augmented set, while evaluation is performed exclusively on the original test partition.

5.1.3 Implementation Details

All experiments are implemented using PyTorch and PyTorch Geometric [41]. All datasets are split using a fixed 80/20 train-test partition with random seed 42 to ensure reproducibility. Results are averaged over 3 independent runs with different random seeds; standard deviations are reported for all metrics.

For the SFFS-based subgraph construction, the default configuration uses a subgraph budget $B = 20$ and a burning probability $p = 0.6$. The labeled anomaly ratio ρ determines the fraction of training anomalies used as seeds, with $|\mathcal{S}| = \lfloor \rho \cdot |\mathcal{A}_{\text{train}}| \rfloor$, where $\mathcal{A}_{\text{train}}$ denotes the anomalous nodes in the training partition.

The GraphSAGE model is trained for 200 epochs with learning rate 0.01 and weight decay 5×10^{-4} , using the Adam optimizer.

5.1.4 Evaluation Protocol

We adopt a standard inductive evaluation protocol. The model is trained exclusively on the training partition $\mathcal{V}_{\text{train}}$ - or its augmented version $\mathcal{V}'_{\text{train}}$ in the SFFS setting. After training, the model assigns an anomaly score $s_i \in [0,1]$ to each node in the test partition $\mathcal{V}_{\text{test}}$.

Performance is evaluated using two standard metrics: Area Under the ROC Curve (AUC-ROC) and Area Under the Precision-Recall Curve (AUC-PR), both

described in Section 2.3.4. These metrics are computed exclusively on the test partition, which remains untouched during training and augmentation.

The baseline is GraphSAGE trained on the original training set $\mathcal{V}_{\text{train}}$ without any SFFS augmentation. The proposed method trains the same model on the augmented training set $\mathcal{V}'_{\text{train}}$, keeping all other settings identical. The difference in performance between the two settings isolates the contribution of the SFFS training augmentation.

5.2 Main Results

Table 5.2 reports the performance of baseline GraphSAGE and GraphSAGE augmented with SFFS at a labeled anomaly ratio of 10% across the six benchmark datasets. The choice of 10% follows the experimental setting adopted in MGAD [1], which serves as one of the main inspirations for the proposed framework. We also studied the effect of varying this ratio in a more few-shot setting in Section 5.3.1.

Results are averaged over three independent runs with standard deviations consistently below 0.005 across all configurations, indicating stable and reproducible results. We report both AUC-ROC and AUC-PR, together with the absolute improvement (Δ) introduced by SFFS augmentation.

Table 5.2: Main results on the six benchmark datasets at labeled ratio 10%. Δ denotes the absolute improvement introduced by SFFS augmentation. Best results are reported in **bold**.

Dataset	AUC-ROC			AUC-PR		
	Baseline	+SFFS	Δ	Baseline	+SFFS	Δ
Cora	0.728	0.763	+0.035	0.221	0.379	+0.157
Citeseer	0.749	0.794	+0.045	0.181	0.314	+0.132
BlogCatalog	0.861	0.890	+0.029	0.306	0.358	+0.052
Flickr	0.829	0.861	+0.032	0.324	0.406	+0.082
ACM	0.854	0.863	+0.009	0.263	0.296	+0.033
Pubmed	0.926	0.944	+0.017	0.477	0.660	+0.183

Overall, SFFS augmentation improves performance on all six datasets. The gains are particularly evident in AUC-PR, which is generally considered the more informative metric in anomaly detection due to the severe class imbalance between anomalous and normal nodes. Across the benchmark, improvements in AUC-PR range from +0.033 on ACM to +0.183 on Pubmed, while AUC-ROC gains are smaller and more variable, ranging from +0.009 on ACM to +0.045 on Citeseer.

The strongest improvements are observed on Pubmed and Cora, where AUC-PR increases by +0.183 and +0.157 respectively. These results suggest that the local

neighborhoods around anomaly seeds contain useful structural information that can be exploited through augmentation.

On ACM, the improvements are more moderate (+0.009 AUC-ROC, +0.033 AUC-PR), while remaining consistently positive. This suggests that the effectiveness of SFFS is not only driven by the quality of the local neighborhoods around anomaly seeds, but also by their relative contribution to the training set. In ACM, the combination of a larger graph size and a fixed subgraph budget reduces the influence of the augmented samples, even in the presence of meaningful structural enrichment.

Citeseer also benefits substantially from SFFS augmentation, achieving the largest AUC-ROC improvement among all datasets (+0.045) together with a strong AUC-PR gain (+0.132). These results indicate that the proposed augmentation strategy remains effective across different graph structures and anomaly distributions.

BlogCatalog and Flickr present a different behavior. For these graphs, which are the densest ones in our benchmark, the SFFS expansion tends to include many normal nodes alongside anomalous ones, reducing the anomaly concentration in the generated subgraphs. Despite this, improvements are still consistently observed on both AUC-ROC and AUC-PR, suggesting robustness to noisier structural contexts and confirming that even a partially enriched augmentation can provide useful signal to the classifier.

Overall, the results indicate that SFFS-based augmentation consistently improves anomaly detection performance across datasets with different structural properties. The magnitude of the improvement varies depending on the interaction between the structural characteristics of the graph and the effectiveness of the generated subgraphs. In particular, the effectiveness of SFFS appears to depend on two complementary factors: the anomaly concentration in the local neighborhoods around the seeds and the relative impact of the augmented subgraphs with respect to the training set. When these conditions are favorable the benefits of the method become more pronounced. The coverage analysis in Section 5.4 helps explain these differences by linking performance to the level of anomaly enrichment in the generated subgraphs.

5.3 Ablation Studies

This section investigates the contribution of the main components and hyperparameters of the proposed SFFS training augmentation framework. The goal of these experiments is to evaluate how the method behaves under different supervision levels and subgraph sampling configurations.

Unless otherwise specified, all ablation studies use the `fe_generic` strategy with default parameters: subgraph budget $B = 20$, burning probability $p = 0.6$, and labeled anomaly ratio $\rho = 10\%$. Experiments are conducted on all six benchmark

datasets and evaluated using both $\Delta\text{AUC-ROC}$ and $\Delta\text{AUC-PR}$ with respect to the baseline GraphSAGE without augmentation. Reported results correspond to the mean over three independent runs; standard deviations are below 0.05 across all configurations.

The ablation analysis focuses on four main aspects:

- **Effect of Labeled Ratio.** We analyze how the amount of available supervision influences performance by varying the percentage of labeled anomalous nodes used as seeds, from 1% to 10%.
- **Effect of Subgraph Budget.** We study the impact of the subgraph budget B , which controls the maximum number of nodes explored during SFFS and determines the size of the generated subgraphs.
- **Effect of Burning Probability.** We evaluate the impact of the burning probability p , which controls the breadth of the subgraph expansion during sampling.
- **Effect of Subgraph Construction Strategy.** We compare four subgraph construction strategies - `fe_generic`, `fe_path`, `fe_schema`, and `fe_wind` - to study how different ways of building anomaly-centered subgraphs affect detection performance across datasets with different structural properties.

5.3.1 Effect of Labeled Ratio

Tables 5.3 and 5.4 report the effect of varying the labeled anomaly ratio ρ on $\Delta\text{AUC-ROC}$ and $\Delta\text{AUC-PR}$ respectively. The ratio represents the percentage of training anomalies used as seeds for subgraph construction. For each value of ρ , the number of seeds is defined as $|\mathcal{S}| = \lfloor \rho \cdot |\mathcal{A}_{\text{train}}| \rfloor$. For example, on Cora, the training split contains approximately 120 anomalous nodes. Therefore, a ratio of $\rho = 1\%$ corresponds to using only one labeled anomaly seed, while $\rho = 10\%$ provides around twelve seeds for subgraph construction.

The results show that SFFS augmentation remains effective across different supervision levels. In general, increasing the labeled anomaly ratio leads to larger improvements, as more seeds allow the method to construct a higher number of anomaly-centered subgraphs and provide the classifier with a richer training context.

The most consistent trend is observed on Pubmed, where the improvement in AUC-PR increases from +0.062 at 1% supervision to +0.183 at 10%. A similar behavior can be observed on Cora, where the largest gain is obtained at the highest ratio (+0.157 AUC-PR). These results suggest that when additional labeled anomalies are available, SFFS can better exploit the local graph structure around anomalous regions.

Table 5.3: Effect of labeled anomaly ratio on $\Delta\text{AUC-ROC}$ (`fe_generic`, $B = 20$, $p = 0.6$). Best per row in **bold**.

Dataset	1%	3%	5%	7%	10%
Cora	+0.031	+0.041	+0.036	+0.035	+0.035
Citeseer	+0.003	+0.006	-0.025	-0.018	+0.045
BlogCatalog	-0.023	-0.012	-0.003	+0.006	+0.029
Flickr	+0.018	+0.014	+0.012	+0.021	+0.032
ACM	+0.004	+0.016	+0.010	+0.006	+0.009
Pubmed	+0.001	+0.007	+0.013	+0.017	+0.017

Table 5.4: Effect of labeled anomaly ratio on $\Delta\text{AUC-PR}$ (`fe_generic`, $B = 20$, $p = 0.6$). Best per row in **bold**.

Dataset	1%	3%	5%	7%	10%
Cora	+0.077	+0.088	+0.090	+0.090	+0.157
Citeseer	-0.013	+0.002	+0.014	+0.007	+0.132
BlogCatalog	-0.009	+0.011	+0.043	+0.039	+0.052
Flickr	+0.020	+0.017	+0.027	+0.027	+0.082
ACM	+0.015	+0.014	+0.020	+0.020	+0.033
Pubmed	+0.062	+0.118	+0.146	+0.173	+0.183

Citeseer presents a less stable behavior at low supervision levels. The AUC-ROC improvement is close to zero or slightly negative for small ratios, while the performance increases significantly at 10% supervision (+0.045 AUC-ROC and +0.132 AUC-PR). This indicates that, in sparse graphs, a small number of seeds may provide limited structural coverage, while additional seeds allow the augmentation process to capture more representative anomaly neighborhoods.

BlogCatalog shows a similar trend, with small negative improvements at 1% and 3% ratios that become positive as more seeds are introduced. Since BlogCatalog is a dense graph, individual subgraphs generated from a small number of seeds may represent only a limited portion of the training distribution. Increasing the number of seeds improves the coverage of the augmented training set.

For Flickr and ACM, improvements remain positive across all ratios, although the magnitude is smaller compared to Pubmed and Cora. This suggests that the method is robust even when the structural signal around the seeds is less pronounced.

Overall, these results show that SFFS augmentation does not require a large amount of labeled anomalies to provide benefits, but its effectiveness generally increases as more supervision becomes available. The impact of the ratio is therefore

related not only to the number of seeds, but also to how representative the generated subgraphs are with respect to the underlying graph structure.

5.3.2 Effect of Subgraph Budget

Tables 5.5 and 5.6 report the effect of varying the subgraph budget $B \in \{4, 10, 20, 30, 50\}$ on $\Delta\text{AUC-ROC}$ and $\Delta\text{AUC-PR}$ respectively, while keeping the other parameters fixed ($\rho = 10\%$, $p = 0.6$).

Table 5.5: Effect of subgraph budget on $\Delta\text{AUC-ROC}$ (`fe_generic`, $\rho = 10\%$, $p = 0.6$). Best per row in **bold**.

Dataset	4	10	20	30	50
Cora	+0.003	+0.034	+0.035	+0.052	+0.090
Citeseer	+0.003	+0.002	+0.045	+0.039	+0.073
BlogCatalog	-0.015	-0.006	+0.029	+0.040	+0.041
Flickr	+0.015	+0.006	+0.032	+0.031	+0.045
ACM	+0.008	+0.023	+0.029	+0.028	+0.031
Pubmed	+0.002	+0.021	+0.017	+0.014	+0.017

Table 5.6: Effect of subgraph budget on $\Delta\text{AUC-PR}$ (`fe_generic`, $\rho = 10\%$, $p = 0.6$). Best per row in **bold**.

Dataset	4	10	20	30	50
Cora	+0.021	+0.073	+0.157	+0.175	+0.258
Citeseer	+0.007	+0.075	+0.132	+0.210	+0.164
BlogCatalog	+0.002	-0.001	+0.052	+0.085	+0.140
Flickr	+0.014	+0.023	+0.082	+0.095	+0.128
ACM	+0.025	+0.112	+0.116	+0.155	+0.174
Pubmed	+0.048	+0.175	+0.184	+0.200	+0.244

The subgraph budget controls the amount of structural context added around each labeled anomaly seed. Small budgets generate compact subgraphs that may capture only a limited portion of the local neighborhood, while larger budgets allow the augmentation process to include more surrounding nodes.

The results show that increasing the budget generally improves AUC-PR, indicating that larger augmented subgraphs provide additional useful information for identifying anomalous nodes under class imbalance. This effect is particularly evident on Cora and Pubmed, where $\Delta\text{AUC-PR}$ reaches +0.258 and +0.244 respectively at $B = 50$.

The behavior of AUC-ROC is more variable. While some datasets benefit from larger budgets, others reach their best performance with smaller subgraphs. For example, on Pubmed the highest AUC-ROC improvement is obtained at $B = 10$ (+0.021), while increasing the budget further leads to similar or slightly lower values. This suggests that very large subgraphs may introduce additional nodes that are less informative for the global ranking measured by AUC-ROC, even though they can still improve the detection of anomalous nodes reflected by AUC-PR.

BlogCatalog shows the largest sensitivity to the budget. With small budgets ($B = 4$ and $B = 10$), the improvement is limited or negative, while larger budgets provide a clear benefit. Since BlogCatalog is a dense graph, very small subgraphs may not capture enough neighborhood information, and a larger context is required before the augmentation becomes effective.

Overall, the results indicate that larger budgets usually provide richer structural information, especially for AUC-PR, but increasing the budget does not always improve AUC-ROC. A budget of $B = 20$ offers a good compromise between performance and computational cost and is therefore used as the default setting in the remaining experiments.

5.3.3 Effect of Burning Probability

Tables 5.7 and 5.8 report the effect of varying the burning probability $p \in \{0.2, 0.4, 0.6, 0.8\}$ on $\Delta\text{AUC-ROC}$ and $\Delta\text{AUC-PR}$ respectively, with all other parameters fixed at their default values ($\rho = 10\%$, $B = 20$).

Table 5.7: Effect of burning probability p on $\Delta\text{AUC-ROC}$ (`fe_generic`, $\rho = 10\%$, $B = 20$). Best per row in **bold**.

Dataset	p=0.2	p=0.4	p=0.6	p=0.8
Cora	+0.038	+0.052	+0.035	+0.037
Citeseer	+0.054	+0.081	+0.045	+0.055
BlogCatalog	+0.029	+0.029	+0.029	+0.029
Flickr	+0.029	+0.034	+0.032	+0.032
ACM	+0.034	+0.023	+0.029	+0.030
Pubmed	+0.017	+0.019	+0.017	+0.019

The burning probability controls the breadth of the SFFS expansion: low values of p produce more chain-like subgraphs that explore deeper paths from the seed, while high values produce wider local neighborhoods. With a fixed budget $B = 20$, both exploration styles capture the same number of nodes, differing only in how they are selected

The results indicate that the burning probability has a relatively limited influence

Table 5.8: Effect of burning probability p on $\Delta\text{AUC-PR}$ (`fe_generic`, $\rho = 10\%$, $B = 20$). Best per row in **bold**.

Dataset	p=0.2	p=0.4	p=0.6	p=0.8
Cora	+0.136	+0.178	+0.157	+0.169
Citeseer	+0.171	+0.173	+0.132	+0.163
BlogCatalog	+0.052	+0.052	+0.052	+0.052
Flickr	+0.074	+0.084	+0.082	+0.082
ACM	+0.148	+0.139	+0.116	+0.120
Pubmed	+0.183	+0.186	+0.184	+0.191

on detection performance. No consistent trend emerges across datasets. On Cora and Citeseer, intermediate values around $p = 0.4$ produce the best results, whereas ACM slightly favors smaller values. On Pubmed, performance remains almost unchanged across the entire range of tested values.

A particularly interesting case is BlogCatalog, where all burning probabilities produce exactly the same results. This behavior suggests that, due to the high density of the graph and the fixed budget ($B=20$), the SFFS the expansion rapidly reaches the maximum allowed size regardless of the burning probability. As a result, the generated subgraphs are very similar and the impact of p becomes negligible since it has little opportunity to alter the composition of the generated subgraphs.

Overall, the results indicate that the burning probability is not a highly sensitive hyperparameter. Although some datasets achieve slightly better performance with $p = 0.4$, the differences are small and no consistent optimal value emerges across datasets.

5.3.4 Effect of Subgraph Construction Strategy

Tables 5.9 and 5.10 report the effect of different subgraph construction strategies on $\Delta\text{AUC-ROC}$ and $\Delta\text{AUC-PR}$ respectively. For schema-guided strategies (`fe_path`, `fe_schema`, `fe_wind`), we report the best result among the three metapath schemas (ANA, ANNA, ANAN).

The results show that unconstrained and schema-guided strategies capture complementary aspects of the graph structure.

The `fe_generic` strategy provides the most stable behaviour across datasets, as it does not require a valid schema path and always produces a subgraph for each seed, making it robust to sparse graphs and low labeled ratios. This makes it robust when the expected anomaly-normal relationships are not consistently present in the local neighborhood.

However, schema-guided strategies can provide additional benefits on datasets

Table 5.9: Effect of subgraph construction strategy on $\Delta\text{AUC-ROC}$ (`fe_generic`, $\rho = 10\%$, $B = 20$, $p = 0.6$). For schema-guided strategies, the best result across ANA, ANNA, ANAN is reported together with the corresponding schema. Best per row in **bold**.

Dataset	<code>fe_generic</code>	<code>fe_path</code>	<code>fe_schema</code>	<code>fe_wind</code>	Best schema
Cora	+0.035	+0.022	+0.035	+0.038	ANNA
Citeseer	+0.045	+0.021	+0.016	+0.001	ANNA
BlogCatalog	+0.029	-0.014	+0.019	+0.017	ANAN
Flickr	+0.032	+0.013	+0.030	+0.028	ANNA
ACM	+0.009	+0.008	+0.031	+0.024	ANNA
Pubmed	+0.017	+0.005	+0.016	+0.015	ANAN

Table 5.10: Effect of subgraph construction strategy on $\Delta\text{AUC-PR}$ (`fe_generic`, $\rho = 10\%$, $B = 20$, $p = 0.6$). For schema-guided strategies, the best result across ANA, ANNA, ANAN is reported together with the corresponding schema. Best per row in **bold**.

Dataset	<code>fe_generic</code>	<code>fe_path</code>	<code>fe_schema</code>	<code>fe_wind</code>	Best schema
Cora	+0.157	+0.038	+0.134	+0.131	ANNA
Citeseer	+0.132	-0.002	+0.020	+0.001	ANNA
BlogCatalog	+0.052	-0.009	+0.052	+0.070	ANA
Flickr	+0.082	+0.030	+0.069	+0.073	ANNA
ACM	+0.033	+0.028	+0.142	+0.132	ANNA
Pubmed	+0.183	+0.071	+0.160	+0.164	ANNA

where specific local patterns are informative. For example, on ACM, `fe_schema` achieves the largest improvement in AUC-PR (+0.142), outperforming `fe_generic` (+0.033). This suggests that enforcing the presence of anomaly-normal structural relations can provide more discriminative information than unconstrained expansion.

Similarly, `fe_wind` often provides competitive performance, combining the structural constraint of schema-guided exploration with the larger neighborhood coverage obtained through SFFS expansion. This is particularly evident on BlogCatalog, Flickr and Pubmed, where `fe_wind` achieves the highest AUC-PR improvements.

On the other hand, `fe_path` consistently underperforms the other strategies. Since it only keeps nodes that belong to a valid schema path, it focuses on satisfying the structural constraint rather than maximizing neighborhood coverage. As a consequence, the resulting subgraphs are smaller and may contain less contextual information for the detector.

The optimal metapath schema also varies across datasets. Although no single schema is universally optimal, ANNA emerges as the most frequently selected pattern, particularly on Cora, Citeseer, Flickr, ACM and Pubmed. In contrast, BlogCatalog tends to favor shorter or alternative anomaly-normal relations. This suggests that the usefulness of a schema depends on the local organization of anomalous and normal nodes, and that different graph structures may benefit from different types of structural constraints.

Overall, the results indicate that no single construction strategy dominates across all datasets. While **fe_generic** represents a reliable default due to its robustness and simplicity, schema-aware strategies can improve performance when the graph contains informative anomaly-normal structural patterns.

5.4 Coverage Analysis

To better understand why SFFS augmentation produces stronger improvements on some datasets than others, we analyze the structural properties of the subgraphs constructed by **fe_generic**. Specifically, we measure three complementary quantities for each dataset across labeled ratios $\rho \in \{1\%, 3\%, 5\%, 7\%, 10\%\}$:

- **Coverage** measures the fraction of unlabeled anomalous nodes that are reached by at least one generated subgraph. Higher coverage means that SFFS is able to discover more anomalous nodes that were not used as seeds during training.
- **Enrichment** measures how concentrated anomalies are inside the generated subgraphs compared to the full graph. Formally, it is the ratio between the anomaly rate inside the subgraphs and the global anomaly rate. For example, if the graph contains 5% anomalies but the generated subgraphs contain 30% anomalies, the enrichment factor is $6\times$, meaning that SFFS is sampling a neighborhood that is six times more anomaly-rich than a random sample of the same size.
- **Overlap** measures how similar the generated subgraphs from different seeds are. Low overlap means that different seeds explore different regions of the graph and provide complementary information instead of repeatedly sampling the same nodes.

Table 5.11 reports the full results, while Figure 5.1 shows the anomaly enrichment trends across datasets and labeled ratios.

The results show that SFFS consistently produces subgraphs that are richer in anomalies than the original graph, with enrichment values above $1\times$ across all datasets. This confirms that the expansion process is not equivalent to random

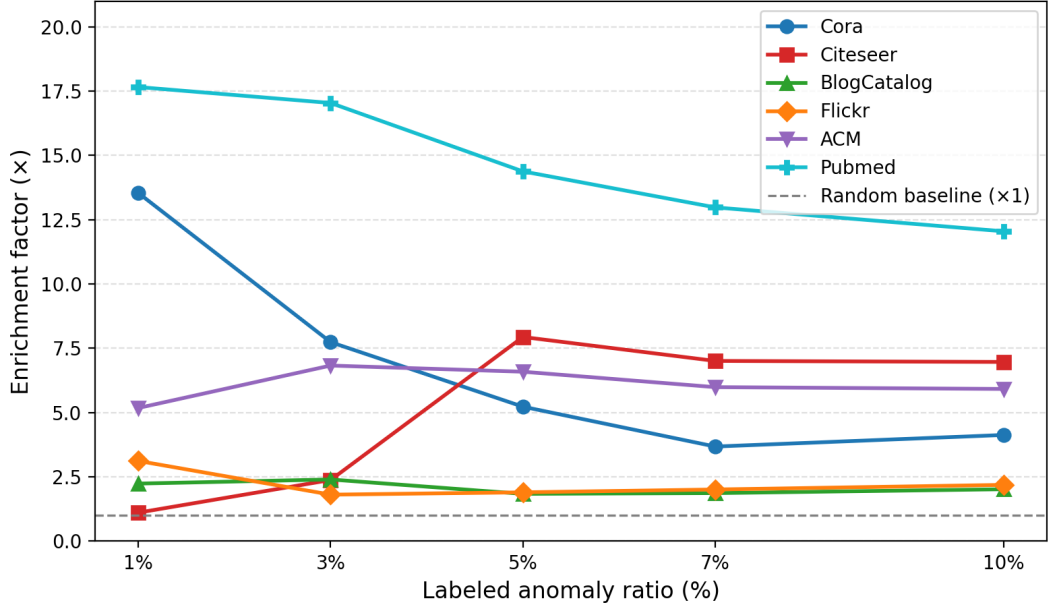


Figure 5.1: Anomaly enrichment of `fe_generic` subgraphs ($B = 20$, $p = 0.6$) across datasets and labeled ratios. Enrichment is defined as the ratio between the anomaly density inside the subgraphs and the anomaly rate in the original graph. The dashed line represents the random baseline ($1\times$).

sampling: starting from labeled anomalous seeds, SFFS tends to reach structurally related anomalous regions. However, higher enrichment does not necessarily correspond to higher labeled ratios. When only few seeds are available, SFFS tends to focus on very local regions around known anomalies, which can result in highly concentrated subgraphs. As more seeds are introduced, the generated subgraphs cover a larger portion of the graph and reach additional anomalous nodes, but they may also include more normal nodes from different regions. Therefore, coverage can increase while enrichment decreases, reflecting a trade-off between discovering more anomalies and maintaining high anomaly concentration.

Pubmed shows the strongest enrichment across all settings, reaching $17.66\times$ at 1% supervision and remaining above $12\times$ at 10%. This strong anomaly concentration explains the large improvements observed in AUC-PR, since the augmented training samples provide a highly informative signal for the classifier.

Cora follows a similar trend, with high enrichment especially at low ratios. When only few seeds are available, they are likely located in dense anomalous regions, producing highly focused subgraphs. Increasing the number of seeds improves coverage but slightly reduces enrichment, as additional regions of the graph are explored.

ACM represents an interesting case. Although its enrichment is higher than

Table 5.11: Coverage analysis of `fe_generic` subgraphs ($B = 20$, $p = 0.6$). Coverage: fraction of unlabeled anomalies reached. Enrichment: anomaly density in subgraphs vs global rate. Overlap: mean Jaccard overlap between subgraph pairs.

Dataset	Metric	Labeled ratio				
		1%	3%	5%	7%	10%
Cora	Coverage	0.094	0.102	0.103	0.105	0.209
	Enrichment	13.54	7.74	5.23	3.68	4.13
	Overlap	0.000	0.000	0.003	0.002	0.009
Citeseer	Coverage	0.000	0.020	0.200	0.210	0.321
	Enrichment	1.11	2.38	7.94	7.01	6.97
	Overlap	0.000	0.037	0.014	0.007	0.004
BlogCatalog	Coverage	0.010	0.041	0.045	0.064	0.102
	Enrichment	2.24	2.40	1.84	1.87	2.02
	Overlap	0.026	0.002	0.001	0.003	0.003
Flickr	Coverage	0.018	0.023	0.040	0.059	0.100
	Enrichment	3.12	1.81	1.90	2.01	2.19
	Overlap	0.000	0.007	0.008	0.007	0.006
ACM	Coverage	0.019	0.094	0.148	0.183	0.255
	Enrichment	5.18	6.83	6.59	5.99	5.92
	Overlap	0.000	0.000	0.003	0.002	0.002
Pubmed	Coverage	0.065	0.169	0.234	0.298	0.361
	Enrichment	17.66	17.04	14.38	12.98	12.05
	Overlap	0.000	0.029	0.019	0.012	0.012

Cora at 10% ($5.92\times$ vs $4.13\times$), the performance gains are smaller. This suggests that enrichment alone does not determine the effectiveness of the augmentation. The impact also depends on how many new nodes are added compared to the size of the graph. Since ACM is much larger than Cora, the same budget $B = 20$ introduces a smaller amount of additional information, reducing the overall benefit of the augmentation.

Citeseer highlights the importance of the number of available seeds. At very low supervision, coverage is limited and enrichment is close to the random baseline. As more labeled anomalies are available, SFFS is able to discover additional anomalous regions, leading to stronger performance improvements.

Finally, BlogCatalog and Flickr show lower enrichment values compared to the other datasets. Due to their denser structure, the expansion process includes more normal nodes together with anomalies, reducing the anomaly concentration

inside the generated subgraphs. Nevertheless, the augmentation still improves performance, showing that even partially enriched neighborhoods can provide useful structural information.

The overlap remains consistently low across datasets, indicating that different seeds generally explore different graph regions and provide complementary information rather than redundant samples.

Overall, the analysis suggests that the effectiveness of SFFS depends on two main factors: the anomaly enrichment of the generated subgraphs and how much the added nodes influence the training set. High enrichment provides more informative samples, while smaller graphs benefit more from the same augmentation budget.

5.5 Complexity Analysis

A key practical advantage of the proposed framework is its very low computational cost. The SFFS augmentation step - which includes subgraph construction and the extension of the training set - is performed only once before training and does not require any gradient-based optimization.

Table 5.12 reports the SFFS augmentation time and the GraphSAGE training time across all datasets, averaged over 3 runs of 200 epochs.

Table 5.12: Computational cost of SFFS augmentation compared to GraphSAGE training time. Each measurement corresponds to a single training run of 200 epochs, averaged over 3 runs. $\rho = 10\%$, $B = 20$, $p = 0.6$.

Dataset	SFFS time (s)	GraphSAGE training (s)
Cora	0.009 ± 0.005	~ 5.3
Citeseer	0.007 ± 0.000	~ 11.8
BlogCatalog	0.078 ± 0.004	~ 259.9
Flickr	0.106 ± 0.013	~ 446.5
ACM	0.081 ± 0.052	~ 122.4
Pubmed	0.027 ± 0.002	~ 7.7

The results show that the computational overhead introduced by SFFS is negligible. Across all datasets, the augmentation step requires less than 0.11 seconds, while GraphSAGE training ranges from approximately 5 seconds to more than 440 seconds. Consequently, the augmentation cost represents only a very small fraction of the total runtime.

Another interesting observation is that the augmentation time does not scale directly with the number of nodes. For example, Pubmed (19,717 nodes) is faster than BlogCatalog (5,196 nodes) and Flickr (7,575 nodes) despite being much larger.

This suggests that the cost of SFFS depends more on the local graph structure around the seeds than on the total graph size. In particular, denser graphs require more neighborhood exploration during the expansion process, leading to slightly higher preprocessing times.

A similar effect can be observed for GraphSAGE training. The longest training times are obtained on BlogCatalog and Flickr, which are also the densest graphs in the benchmark. In contrast, Pubmed requires only 7.7 seconds despite being the largest graph. This indicates that graph density has a stronger impact on the computational cost than the number of nodes alone.

Overall, the results confirm that SFFS augmentation adds negligible computational overhead to the anomaly detection pipeline while providing consistent performance gains. This makes the proposed framework a practical and efficient solution for graph anomaly detection with very few labels.

5.6 Discussion

We summarize the main findings of the experimental evaluation and discuss what they reveal about the behavior of the proposed SFFS training augmentation framework.

SFFS augmentation consistently improves detection performance. Across all six datasets and five labeled ratios, SFFS augmentation consistently improves anomaly detection performance. This result is particularly exceptional because the datasets have very different structural characteristics, including graph size, density, feature dimensionality, and anomaly rate. Since the method only modifies the training set without changing the classifier architecture, it can be easily integrated into existing graph anomaly detection pipelines.

AUC-PR is more informative in the few-shot setting. AUC-PR provides a more informative view of the improvements in this setting due to the strong class imbalance between normal and anomalous nodes. Across several datasets, AUC-ROC improvements are relatively small, while AUC-PR shows larger gains. This suggests that SFFS augmentation is particularly effective at improving the ranking of anomalous nodes among the most suspicious predictions, which is more relevant when anomalies are rare.

The quality of the subgraphs plays a key role. The largest improvements are observed on Pubmed and Cora, where the coverage analysis reveals the highest anomaly enrichment values. This means that SFFS naturally reaches additional anomalous nodes when expanding from labeled seeds, providing GraphSAGE with a

more informative training context. The effectiveness of the augmentation therefore depends not only on the number of seeds, but also on how anomaly-rich the local neighborhoods are.

Coverage and enrichment reveal a trade-off. The coverage analysis shows that increasing the number of labeled seeds generally improves coverage, allowing SFFS to discover a larger fraction of unlabeled anomalies. However, higher coverage does not always correspond to higher enrichment. As more seeds are added, the generated subgraphs explore more diverse graph regions and may include a larger number of normal nodes. This creates a natural trade-off between discovering more anomalies and maintaining highly concentrated anomaly-rich neighborhoods.

Graph size interacts with augmentation impact. The relative contribution of the augmented nodes to the training set also influences performance. On smaller graphs like Cora, the nodes added by SFFS represent a larger fraction of the training set, amplifying the impact of the augmentation. On larger graphs like ACM, the same budget $B = 20$ adds proportionally fewer nodes, limiting the relative benefit even when enrichment is substantial.

fe_generic is the most reliable construction strategy. Among the tested subgraph construction strategies, **fe_generic** provides the most stable behavior across datasets. It never fails during subgraph construction and consistently achieves competitive results. Schema-guided strategies such as **fe_schema** and **fe_wind** can outperform **fe_generic** on specific datasets - particularly ACM - but their effectiveness depends on the local graph structure supporting the required metapath patterns. When prior knowledge about the local anomaly structure is available, schema-guided strategies may therefore be preferred over **fe_generic**.

The burning probability is a robust hyperparameter. The ablation study on the burning probability shows that performance is largely insensitive to the value of p . No consistent optimal value emerges across datasets, and the differences between values are small. This confirms that p does not require careful tuning and that the default value $p = 0.6$ is a reliable choice.

Larger budgets generally improve AUC-PR. Increasing the subgraph budget B consistently improves AUC-PR across all datasets, as larger subgraphs capture more structural context around the anomaly seeds. The effect on AUC-ROC is more variable, suggesting that additional nodes help identify anomalies more precisely but do not always improve the global ranking performance.

The method remains effective with very few labels. Even at a labeled ratio of 1% - corresponding to as few as 1 labeled seed on some datasets - SFFS augmentation produces meaningful improvements on most datasets. This confirms that the method can exploit very limited supervision and remains effective in realistic anomaly detection scenarios where labeled anomalies are scarce and expensive to obtain.

The generated subgraphs provide complementary information. The overlap analysis shows consistently low overlap values across all datasets and labeled ratios. This indicates that different seeds tend to explore different regions of the graph instead of repeatedly sampling the same nodes. As a result, the augmented nodes provide complementary structural information rather than redundant training examples.

The method is computationally efficient. The complexity analysis shows that SFFS augmentation introduces negligible computational overhead. The augmentation step always requires less than 0.11 seconds and is executed only once before training, while GraphSAGE training requires from several seconds to several minutes depending on the dataset. This makes the proposed framework practical and scalable even for large graphs

Chapter 6

Conclusion

This thesis investigated the problem of graph anomaly detection under very few labels, a realistic setting in which only a limited number of anomalous nodes are available during training. In such scenarios, the main challenge is to exploit the scarce supervision effectively while still providing the model with enough structural information to identify previously unseen anomalies.

To address this problem, we proposed a training augmentation framework based on Subgraph Forest Fire Sampling (SFFS). Starting from a small set of labeled anomaly seeds, SFFS constructs local subgraphs around the seeds and augments the training set with the discovered nodes. This allows the classifier to learn from a richer structural context without modifying the underlying model architecture.

The proposed framework was evaluated on six benchmark datasets covering citation and social networks with different sizes, densities, and anomaly distributions. The experimental results show that SFFS augmentation consistently improves anomaly detection performance while introducing negligible computational overhead.

This chapter summarizes the main contributions of the thesis, discusses the principal findings of the experimental evaluation, highlights the limitations of the proposed approach, and outlines possible directions for future work.

6.1 Summary of Contributions

The main contributions of this thesis can be summarized as follows:

- **A training augmentation framework for few-label graph anomaly detection.** The proposed method enriches the training set with nodes structurally close to known anomalies. The framework operates as a preprocessing step and can be integrated with graph classifiers without modifying their architecture.

- **An SFFS-based subgraph construction methodology.** The proposed framework leverages Subgraph Forest Fire Sampling to explore the local neighborhoods around labeled anomaly seeds and generate anomaly-centered training contexts.
- **An extensive analysis of subgraph construction strategies.** We investigated and systematically compared four variants of SFFS-based subgraph construction - free stochastic expansion (`fe_generic`), path-only (`fe_path`), schema-guided (`fe_schema`), and wind-expanded (`fe_wind`) - covering a spectrum from unconstrained to structurally constrained approaches.
- **A structural coverage analysis.** We introduced a quantitative and structural analysis based on coverage, enrichment, and overlap to better understand the properties of the generated subgraphs and explain the differences observed across datasets.
- **A comprehensive experimental evaluation.** We evaluated the framework on six benchmark datasets across multiple labeled ratios and hyperparameter settings, including ablation studies on the subgraph budget, burning probability, and construction strategy.

6.2 Main Findings

The experimental evaluation reveals several consistent findings.

SFFS augmentation consistently improves detection performance. Across all six benchmark datasets, SFFS augmentation improves anomaly detection performance compared to training on the original data alone. The strongest gains are observed on Pubmed and Cora, where AUC-PR increases by up to +0.183 and +0.157 respectively. These results demonstrate that structural neighborhood augmentation can effectively exploit limited anomaly supervision.

AUC-PR is the most informative metric in this setting. Due to the strong class imbalance inherent to graph anomaly detection, AUC-PR provides a more informative assessment than AUC-ROC. Across multiple datasets, AUC-PR improvements are substantially larger than AUC-ROC gains, indicating that SFFS is particularly effective at improving the ranking of anomalous nodes among the most suspicious predictions.

Coverage and enrichment explain the effectiveness of the augmentation. The coverage analysis shows that the quality of the generated subgraphs is a key

factor in determining performance improvements. High enrichment indicates that the generated subgraphs contain a higher concentration of anomalies than the original graph, while high coverage allows SFFS to reach additional anomalous nodes that were not used as seeds. The interaction between these two factors largely explains the differences observed across datasets.

Graph size influences the impact of augmentation. The effectiveness of SFFS depends not only on the quality of the generated subgraphs but also on their relative contribution to the training set. On smaller graphs, the nodes added by SFFS represent a larger fraction of the available training data, making the augmentation more influential. On larger graphs, the same augmentation budget has a smaller relative impact.

fe_generic is the most robust construction strategy. Among the evaluated strategies, **fe_generic** provides the most stable performance across datasets. It consistently generates useful subgraphs without requiring schema information or specific structural assumptions. Schema-guided strategies can outperform **fe_generic** on particular datasets, but their effectiveness depends on the presence of suitable local structural patterns.

The method remains effective with very few labels. Even in few-shot settings, SFFS augmentation often produces measurable improvements. This suggests that useful structural information can be extracted even from very limited anomaly supervision.

The method is computationally efficient. The complexity analysis shows that the augmentation step requires at most a fraction of a second and is executed only once before training. Compared to GraphSAGE training times, the additional computational cost is negligible, making the framework practical even for larger graphs.

6.3 Limitations

Despite its effectiveness, the proposed framework presents several limitations.

Dependence on labeled anomaly seeds. The method requires at least one labeled anomalous node to construct subgraphs. Consequently, it cannot be directly applied in fully unsupervised settings. Moreover, when very few seeds are available, particularly in sparse graphs, the generated subgraphs may contain limited useful information.

Dependence on local graph structure. The effectiveness of SFFS depends on the quality of the local neighborhoods around the labeled anomaly seeds. If the generated subgraphs contain informative structural patterns, the augmentation can significantly improve detection performance. However, when anomalies are isolated or their neighborhoods provide little useful information, the impact of the augmentation may be more limited.

Synthetic anomaly benchmarks. The datasets used in the experimental evaluation contain anomalies generated through standard anomaly injection procedures. Although these benchmarks are widely adopted in the literature, the behavior of the method on datasets containing naturally occurring anomalies may differ.

Fixed augmentation budget. The current framework uses a fixed subgraph budget across all datasets and seeds. As graph size increases, the relative contribution of the augmented nodes decreases, potentially limiting the impact of the augmentation on very large graphs.

6.4 Future Directions

Several directions for future research emerge from this work.

Evaluation on organic anomaly datasets. A natural next step is to evaluate the framework on datasets containing naturally occurring anomalies rather than synthetically injected ones. Such experiments would provide additional evidence regarding the applicability of the method in real-world scenarios.

Improved seed selection strategies. The current implementation samples anomaly seeds uniformly at random from the labeled anomalies. A more informed approach could select seeds that maximize coverage, enrichment, or structural diversity, potentially generating more informative subgraphs.

Adaptive subgraph construction. Future work could investigate adaptive versions of SFFS that automatically adjust the subgraph budget or expansion behavior according to local graph characteristics. This could improve robustness across graphs with different densities and structural properties.

Extension to heterogeneous graphs. The current framework operates on homogeneous graphs. Extending it to heterogeneous graphs would allow the use of richer metapath definitions that capture domain-specific relational patterns, opening new application domains such as knowledge graphs and biomedical networks.

Automatic schema discovery. Future work could explore a wider range of meta-path schemas of different lengths and structures, or design data-driven approaches that automatically select the most informative schema for a given dataset.

Weakly supervised subgraph construction. The strongest limitation of the current framework is its dependence on labeled anomaly seeds. Future work could investigate the use of weak supervision, pseudo-labels, or self-supervised objectives to guide subgraph construction, reducing the reliance on labeled anomaly seeds.

Integration with advanced graph anomaly detectors. GraphSAGE was used throughout this thesis to isolate the effect of the proposed augmentation strategy. Future work could investigate whether more expressive architectures - such as attention-based GNNs or detectors with separate structure and attribute encoders - would benefit from the SFFS augmentation.

6.5 Closing Remarks

In conclusion, the results of this thesis show that structural neighborhood augmentation represents a simple and effective way of exploiting limited anomaly supervision. By enriching the training set with nodes that are structurally related to known anomalies, SFFS provides additional context that improves detection performance while maintaining low computational cost. These findings suggest that subgraph-based augmentation is a promising direction for few-label graph anomaly detection and deserves further investigation in more realistic and challenging settings.

Bibliography

- [1] Hwan Kim, Junghoon Kim, Byung Suk Lee, and Sungsu Lim. *Label-based Graph Augmentation with Metapath for Graph Anomaly Detection*. Apr. 12, 2024. DOI: 10.48550/arXiv.2308.10918. arXiv: 2308.10918[cs]. URL: <http://arxiv.org/abs/2308.10918> (visited on 12/23/2025) (cit. on pp. 1, 21, 23, 25, 36).
- [2] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. «Graph neural networks: A review of methods and applications». In: *AI Open* 1 (2020), pp. 57–81. ISSN: 2666-6510. DOI: <https://doi.org/10.1016/j.aiopen.2021.01.001>. URL: <https://www.sciencedirect.com/science/article/pii/S2666651021000012> (cit. on p. 5).
- [3] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. *Neural Message Passing for Quantum Chemistry*. 2017. arXiv: 1704.01212 [cs.LG]. URL: <https://arxiv.org/abs/1704.01212> (cit. on p. 5).
- [4] Qimai Li, Zhichao Han, and Xiao-Ming Wu. «Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning». In: *arXiv preprint arXiv:1801.07606* (2018). URL: <https://arxiv.org/abs/1801.07606> (cit. on p. 6).
- [5] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. «Deep Anomaly Detection on Attributed Networks». In: *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*. 2019, pp. 594–602. DOI: 10.1137/1.9781611975673.67 (cit. on pp. 7, 34).
- [6] Haoyi Fan, Fengbin Zhang, and Zuoyong Li. «AnomalyDAE: Dual Autoencoder for Anomaly Detection on Attributed Networks». In: *arXiv preprint arXiv:2002.03665* (2020). URL: <https://arxiv.org/abs/2002.03665> (cit. on p. 7).

- [7] William L. Hamilton, Rex Ying, and Jure Leskovec. «Inductive Representation Learning on Large Graphs». In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017. URL: <https://arxiv.org/abs/1706.02216> (cit. on pp. 8, 28, 34).
- [8] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. «Graph Attention Networks». In: *International Conference on Learning Representations (ICLR)*. 2018. URL: <https://arxiv.org/abs/1710.10903> (cit. on p. 8).
- [9] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. «How Powerful are Graph Neural Networks?» In: *International Conference on Learning Representations (ICLR)*. 2019. URL: <https://arxiv.org/abs/1810.00826> (cit. on p. 9).
- [10] Leman Akoglu, Hanghang Tong, and Danai Koutra. «Graph-based Anomaly Detection and Description: A Survey». In: *Data Mining and Knowledge Discovery* 29.3 (2015), pp. 626–688. DOI: 10.1007/s10618-014-0365-y (cit. on p. 11).
- [11] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, Hui Xiong, and Leman Akoglu. «A Comprehensive Survey on Graph Anomaly Detection with Deep Learning». In: *IEEE Transactions on Knowledge and Data Engineering* (2021). DOI: 10.1109/TKDE.2021.3118815 (cit. on p. 11).
- [12] Hezhe Qiao, Hanghang Tong, Bo An, Irwin King, Charu Aggarwal, and Guan-song Pang. *Deep Graph Anomaly Detection: A Survey and New Perspectives*. 2025. arXiv: 2409.09957 [cs.LG]. URL: <https://arxiv.org/abs/2409.09957> (cit. on p. 11).
- [13] Chunjing Xiao, Jiahui Lu, Xovee Xu, Fan Zhou, Tianshu Xie, Wei Lu, and Lifeng Xu. «Reconciling Attribute and Structural Anomalies for Improved Graph Anomaly Detection». In: *IEEE Transactions on Neural Networks and Learning Systems* 36.9 (2025), pp. 16661–16674. ISSN: 2162-2388. DOI: 10.1109/tnnls.2025.3561172. URL: <http://dx.doi.org/10.1109/TNNLS.2025.3561172> (cit. on p. 11).
- [14] Sunwoo Kim, Soo Yong Lee, Fanchen Bu, Shinhwan Kang, Kyungho Kim, Jaemin Yoo, and Kijung Shin. *Rethinking Reconstruction-based Graph-Level Anomaly Detection: Limitations and a Simple Remedy*. 2024. arXiv: 2410.20366 [cs.LG]. URL: <https://arxiv.org/abs/2410.20366> (cit. on p. 13).
- [15] Alex Howe, Dale Peasley, and Mauricio Papa. «Graph Autoencoders for Detecting Anomalous Intrusions in OT Networks Through Dynamic Link Detection». In: *Proceedings of the IEEE 21st Consumer Communications & Networking Conference (CCNC)*. 2024, pp. 1–6. DOI: 10.1109/CCNC51664.2024.10454841 (cit. on p. 13).

- [16] Thomas N. Kipf and Max Welling. *Variational Graph Auto-Encoders*. 2016. arXiv: 1611.07308 [stat.ML]. URL: <https://arxiv.org/abs/1611.07308> (cit. on p. 13).
- [17] Tom Fawcett. «An introduction to ROC analysis». In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861–874. ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2005.10.010>. URL: <https://www.sciencedirect.com/science/article/pii/S016786550500303X> (cit. on p. 15).
- [18] Jesse Davis and Mark Goadrich. «The Relationship Between Precision-Recall and ROC Curves». In: *Proceedings of the 23rd International Conference on Machine Learning (ICML)*. Pittsburgh, PA, USA, 2006. DOI: 10.1145/1143844.1143874 (cit. on p. 15).
- [19] Takaya Saito and Marc Rehmsmeier. «The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets». In: *PLOS ONE* 10.3 (2015), e0118432. DOI: 10.1371/journal.pone.0118432. URL: <https://doi.org/10.1371/journal.pone.0118432> (cit. on p. 15).
- [20] Chaoqun Cui and Caiyan Jia. *Towards Real-World Rumor Detection: Anomaly Detection Framework with Graph Supervised Contrastive Learning*. 2025. arXiv: 2508.07205 [cs.SI]. URL: <https://arxiv.org/abs/2508.07205> (cit. on pp. 19, 25).
- [21] Hengji Dong, Jing Zhao, Hongwei Yang, Hui He, Jun Zhou, Yunqing Feng, Yangyiye Jin, Rui Liu, and Mengge Wang. «Semi-supervised Graph Anomaly Detection via Multi-view Contrastive Learning». In: *2024 International Joint Conference on Neural Networks (IJCNN)*. 2024 International Joint Conference on Neural Networks (IJCNN). June 2024, pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650001. URL: <https://ieeexplore.ieee.org/document/10650001/> (visited on 12/23/2025) (cit. on pp. 19, 25).
- [22] Fan Xu, Nan Wang, Xuezhi Wen, Meiqi Gao, Chaoqun Guo, and Xibin Zhao. *Few-shot Message-Enhanced Contrastive Learning for Graph Anomaly Detection*. 2023. arXiv: 2311.10370 [cs.LG]. URL: <https://arxiv.org/abs/2311.10370> (cit. on pp. 19, 22, 25).
- [23] Pengbo Li, Hang Yu, and Xiangfeng Luo. «Context-aware graph neural network for graph-based fraud detection with extremely limited labels». In: *AAAI'25/IAAI'25/EAAI'25*. AAAI Press, 2025. ISBN: 978-1-57735-897-8. DOI: 10.1609/aaai.v39i11.33319. URL: <https://doi.org/10.1609/aaai.v39i11.33319> (cit. on pp. 19, 23, 25).

- [24] Tristan Bilot, Nour El Madhoun, Khaldoun Al Agha, and Anis Zouaoui. «Few Edges Are Enough: Few-Shot Network Attack Detection with Graph Neural Networks». In: vol. 14977. 2024, pp. 257–276. DOI: 10.1007/978-981-97-7737-2_15. arXiv: 2501.16964[cs]. URL: <http://arxiv.org/abs/2501.16964> (visited on 12/01/2025) (cit. on pp. 19, 25).
- [25] Liting Li, Yumeng Wang, and Yueheng Sun. *Meta-Learning Based Few-Shot Graph-Level Anomaly Detection*. Oct. 9, 2025. DOI: 10.48550/arXiv.2510.07847. arXiv: 2510.07847[cs]. URL: <http://arxiv.org/abs/2510.07847> (visited on 11/30/2025) (cit. on pp. 20, 25).
- [26] Xiong Xiao Xu, Kaize Ding, Canyu Chen, and Kai Shu. *MetaGAD: Meta Representation Adaptation for Few-Shot Graph Anomaly Detection*. Aug. 23, 2024. DOI: 10.48550/arXiv.2305.10668. arXiv: 2305.10668[cs]. URL: <http://arxiv.org/abs/2305.10668> (visited on 12/23/2025) (cit. on pp. 20, 23, 25).
- [27] Ximing Li, Linghui Li, Xiaoyong Li, Binsi Cai, Jia Jia, Yali Gao, and Shui Yu. «GMFITD: Graph Meta-Learning for Effective Few-Shot Insider Threat Detection». In: *IEEE Transactions on Information Forensics and Security* 19 (2024), pp. 7161–7175. ISSN: 1556-6021. DOI: 10.1109/TIFS.2024.3430106. URL: <https://ieeexplore.ieee.org/document/10601241> (visited on 12/23/2025) (cit. on pp. 20, 25).
- [28] Shiyu Yang. «DARGAT-GDN: A Robust and Adaptive Graph Neural Network for Few-shot Graph Anomaly Detection». In: *2025 5th International Symposium on Computer Technology and Information Science (ISCTIS)*. 2025 5th International Symposium on Computer Technology and Information Science (ISCTIS). May 2025, pp. 190–199. DOI: 10.1109/ISCTIS65944.2025.11065418. URL: <https://ieeexplore.ieee.org/document/11065418> (visited on 12/01/2025) (cit. on pp. 20, 25).
- [29] Jingyan Chen, Guanghui Zhu, Guansong Pang, Chunfeng Yuan, and Yihua Huang. «AffinityTune: A Prompt-Tuning Framework for Few-Shot Anomaly Detection on Graphs». In: *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. KDD '25: The 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Toronto ON Canada: ACM, Aug. 3, 2025, pp. 179–190. ISBN: 979-8-4007-1454-2. DOI: 10.1145/3711896.3736836. URL: <https://dl.acm.org/doi/10.1145/3711896.3736836> (visited on 12/27/2025) (cit. on pp. 21, 25).
- [30] Jiazhen Chen, Sichao Fu, Zhibin Zhang, Zheng Ma, Mingbin Feng, Tony S. Wirjanto, and Qinmu Peng. «Towards Cross-Domain Few-Shot Graph Anomaly Detection». In: *2024 IEEE International Conference on Data Mining (ICDM)*. 2024 IEEE International Conference on Data Mining (ICDM). ISSN:

- 2374-8486. Dec. 2024, pp. 51–60. DOI: 10.1109/ICDM59182.2024.00012. URL: <https://ieeexplore.ieee.org/document/10884324> (visited on 12/01/2025) (cit. on pp. 21, 25).
- [31] Shuo Liu, Di Yao, Lanting Fang, Zhetao Li, Wenbin Li, Kaiyu Feng, Xiaowen Ji, and Jingping Bi. «AnomalyLLM: Few-Shot Anomaly Edge Detection for Dynamic Graphs Using Large Language Models». In: *2024 IEEE International Conference on Data Mining (ICDM)*. 2024 IEEE International Conference on Data Mining (ICDM). ISSN: 2374-8486. Dec. 2024, pp. 785–790. DOI: 10.1109/ICDM59182.2024.00093. URL: <https://ieeexplore.ieee.org/document/10884409> (visited on 12/01/2025) (cit. on pp. 21, 25).
- [32] Hezhe Qiao, Chaoxi Niu, Ling Chen, and Guansong Pang. *AnomalyGFM: Graph Foundation Model for Zero/Few-shot Anomaly Detection*. June 1, 2025. DOI: 10.48550/arXiv.2502.09254. arXiv: 2502.09254[cs]. URL: <http://arxiv.org/abs/2502.09254> (visited on 12/01/2025) (cit. on pp. 21, 25).
- [33] Andrew Shields, Pat Doody, and Robert Sheehy. «EnGraph: Ensemble Based Augmentation for Graph Anomaly Detection». In: *Proceedings of the 4th International Workshop on Large-Scale Graph Data Analytics (LSGDA 2025)*. VLDB 2025 Workshop. 2025 (cit. on pp. 21, 25).
- [34] Ranran Wang, Yin Zhang, and Limei Peng. «Anomaly Detection Service for Blockchain Transactions Using Minimal Substitution-Based Label Propagation». In: *IEEE Transactions on Services Computing* 17.5 (Sept. 2024), pp. 2054–2066. ISSN: 1939-1374. DOI: 10.1109/TSC.2024.3407601. URL: <https://ieeexplore.ieee.org/document/10542363/> (visited on 12/23/2025) (cit. on pp. 21, 25).
- [35] Xiaoxiao Ma, Ruikun Li, Fanzhen Liu, Kaize Ding, Jian Yang, and Jia Wu. «Graph Anomaly Detection with Few Labels: A Data-Centric Approach». In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '24: The 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Barcelona Spain: ACM, Aug. 25, 2024, pp. 2153–2164. ISBN: 979-8-4007-0490-1. DOI: 10.1145/3637528.3671929. URL: <https://dl.acm.org/doi/10.1145/3637528.3671929> (visited on 11/26/2025) (cit. on pp. 22, 25).
- [36] Xujia Li and Lei Chen. «Graph Anomaly Detection with Domain-Agnostic Pre-Training and Few-Shot Adaptation». In: *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 2024 IEEE 40th International Conference on Data Engineering (ICDE). May 2024, pp. 2667–2680. DOI: 10.1109/ICDE60146.2024.00209. URL: <https://ieeexplore.ieee.org/document/10597726/> (visited on 12/23/2025) (cit. on pp. 22, 25).

- [37] Zhu Wang, Shuang Zhou, Junnan Dong, Chang Yang, Xiao Huang, and Shengjie Zhao. *Graph Anomaly Detection with Noisy Labels by Reinforcement Learning*. July 8, 2024. DOI: 10.48550/arXiv.2407.05934. arXiv: 2407.05934[cs]. URL: <http://arxiv.org/abs/2407.05934> (visited on 12/23/2025) (cit. on pp. 22, 23, 25).
- [38] Xuan Li, Dejie Cheng, Luheng Zhang, Chengfang Zhang, and Ziliang Feng. «Few-Shot Graph Anomaly Detection via Dual-Level Knowledge Distillation». In: *Entropy* 27.1 (Jan. 2025). Publisher: Multidisciplinary Digital Publishing Institute, p. 28. ISSN: 1099-4300. DOI: 10.3390/e27010028. URL: <https://www.mdpi.com/1099-4300/27/1/28> (visited on 11/29/2025) (cit. on pp. 22, 25).
- [39] Chelsea Finn, Pieter Abbeel, and Sergey Levine. «Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks». In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Vol. 70. PMLR. 2017, pp. 1126–1135. URL: <https://arxiv.org/abs/1703.03400> (cit. on p. 20).
- [40] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. *Graph Evolution: Densification and Shrinking Diameters*. 2007. arXiv: physics/0603229 [physics.soc-ph]. URL: <https://arxiv.org/abs/physics/0603229> (cit. on pp. 27–29).
- [41] Adam Paszke et al. «PyTorch: An Imperative Style, High-Performance Deep Learning Library». In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf (cit. on p. 35).