IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 June 2026, accepted 17 July 2026, date of publication 27 July 2026, date of current version 10 August 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3717452

SURVEY

Conceptual Relationship Between Classical Machine Learning and Graph Neural Networks in Out-of-Distribution Detection: A Comprehensive Survey

MAO NGUYEN, THIEN PHAM, LONG S. T. NGUYEN, (Student Member, IEEE), DANG LE, TRUNG M. PHAM, HIEU M. PHAM, DUC Q. NGUYEN, XINH LE, ANH H. HUYNH, TRIET T. M. LE, AND THO T. QUAN

URA Research Group, Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City 700000, Vietnam

Viet Nam National University Ho Chi Minh City, Ho Chi Minh City 700000, Vietnam

Corresponding author: Tho T. Quan (qttho@hcmut.edu.vn)

This work was supported by the Murata Science Foundation under Grant 24VH09.

ABSTRACT Out-of-Distribution (OOD) detection, an important branch of Anomaly Detection (AD), aims to recognize inputs that fall outside the training distribution. Existing OOD methods based on statistical modeling, distance measures, one-class classification, reconstruction, or energy-based scoring provide strong foundations, but they can be difficult to apply directly to high-dimensional, unstructured, or relational data. Recent advances in graph-based deep learning, particularly Graph Neural Networks (GNNs), create new opportunities by modeling local and global dependencies among interdependent samples. This survey provides a comprehensive overview of OOD detection methods, organizing them into traditional and deep learning families before examining graph-aware GNN-based approaches. Beyond taxonomy, we synthesize the conceptual relationships between these paradigms through three working hypotheses: graph-aware transfer, structure-sensitive scoring, and hybrid objective design. This hypothesis-guided synthesis characterizes how established statistical, geometric, reconstruction-based, and energy-based principles can be transferred, reformulated, and combined within modern GNN frameworks. To make this perspective concrete, we include Graph Energy-based OOD Detection (GEO), a reference case study that combines energy-based scoring with one-class regularization in a GNN architecture and reports competitive but setting-dependent performance across representative benchmarks. Taken together, the survey clarifies the current landscape of OOD detection and highlights how integrating classical statistical reasoning with graph-based learning can inform future graph OOD research. To support reproducibility, the implementation of GEO is publicly available at https://github.com/longstnguyen/GEO.

INDEX TERMS Out-of-distribution detection, anomaly detection, graph neural networks, deep learning, statistical methods, machine learning survey, safety-critical applications.

I. INTRODUCTION

Anomaly Detection (AD) refers to the task of identifying patterns, behaviors, or data instances that deviate from what is considered normal. It serves as a core component of robust *Artificial Intelligence* (AI) systems, particularly in safety-critical domains such as healthcare, autonomous driving, and financial services [1], [2], [3]. Within this broader field, *Out-of-Distribution* (OOD) detection has emerged as a specialized subtask. Unlike general AD, which focuses on deviations within a known distribution, OOD detection aims to recognize inputs originating from distributions unseen

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

during training [4]. This distinction is important for reliability and safety when AI systems are exposed to novel or unforeseen conditions.

Despite extensive research, traditional OOD detection methods, typically grounded in statistical modeling, density estimation, or distance-based metrics, exhibit notable limitations. These approaches often assume structured and low-dimensional data, making them less effective in generalizing to complex, relational, or high-dimensional scenarios [5], [6]. When exposed to OOD inputs, such models may yield overconfident or misleading predictions, undermining trust in critical applications such as medical diagnosis and autonomous driving [1], [2].

Recent advances in *Deep Learning* (DL), particularly the emergence of *Graph Neural Networks* (GNNs), offer new opportunities to address these challenges [7]. GNNs are designed to operate directly on graph-structured data, capturing both local and global dependencies among entities by propagating and aggregating information across nodes and their neighbors. This capability enables GNNs to learn context-aware representations for structural and contextual anomalies, positioning them as a relevant framework for OOD detection in complex, interconnected data environments [8].

Existing survey papers have reviewed related literature from several important but different perspectives. Pang et al. [9] survey deep AD broadly and organize the field through a comprehensive taxonomy of deep models. Lu et al. [10] review recent advances in OOD detection from a task-oriented perspective, categorizing methods according to the user's access to the model and whether the detector can modify or retrain it. Qiao et al. [11] summarize deep graph AD from three methodological perspectives, namely GNN backbone design, proxy task design, and graph anomaly measures. These surveys provide valuable overviews of their respective areas. Our survey complements them by centering the *conceptual relationship* between classical non-graph OOD principles and their graph-based reinterpretations. Its contribution lies in consolidating established ideas such as one-class classification, density estimation, distance-based scoring, and energy-based scoring into a transfer-oriented view that makes explicit (i) which classical assumption or score is being reused, (ii) how the score is embedded into a GNN representation space, and (iii) what graph-specific inductive bias is added through message passing, topology-aware propagation, or spectral filtering. This view explains how foundational OOD principles can be systematically reinterpreted within modern GNN-based detection frameworks.

Motivated by these developments, this survey aims to provide not only a comprehensive taxonomy of OOD detection methods but also a principled view of the conceptual connections between classical approaches and emerging GNN-based techniques. By organizing these "conceptual bridges," we seek to clarify how the statistical foundations of classical methods can be reformulated within graph-based learning frameworks and how recent graph learning advances, including uncertainty-aware GNNs, graph energy models, test-time graph OOD methods, heterophily-aware framelets, and spectral high-pass designs, enlarge the design space for graph OOD detection [12], [13], [14], [15], [16], [17]. To illustrate this perspective, we include *Graph Energy-based Out-of-Distribution Detection* (GEO) as a reference case study that integrates energy-based scoring and one-class regularization within a GNN and demonstrates how the proposed conceptual relationships can be instantiated in practice.

Working hypotheses of this survey.

- **H1: Graph-aware transfer.** Classical OOD principles such as density estimation, distance-based scoring, one-class classification, reconstruction error, and energy scoring can be transferred to graph settings by applying them to GNN-learned representations or graph-conditioned logits.
- **H2: Structure-sensitive scoring.** Effective graph OOD detection should not treat samples as independent points only; it should also exploit graph-specific information such as neighborhood topology, message propagation, spectral behavior, heterophily, temporal shift, and higher-order relations.
- **H3: Hybrid objective design.** Combining complementary classical principles within GNN frameworks may provide richer graph OOD evidence than relying on a single score or objective, especially when density, geometry, and relational structure provide different signals.

The main contributions of this paper are summarized as follows.

- We delineate the conceptual and operational differences between AD and OOD detection, establishing the latter as the central focus of this review.
- We provide a comprehensive taxonomy of classical OOD detection methods, including probabilistic models, distance-based techniques, and one-class classification approaches, while outlining their respective strengths and limitations.
- We systematically review DL-based OOD detection methods, with emphasis on recent advances in graph-based learning, covering reconstruction-based models, generative models, gradient-based techniques, and various GNN architectures.
- We consolidate and analyze the conceptual relationships between classical and GNN-based methods, highlighting how principles such as energy scoring and one-class classification can be reinterpreted or extended within GNN frameworks. These connections are organized into a transfer-oriented perspective that specifies the reused classical score, the learned graph representation, and the graph-specific inductive bias.
- We instantiate these insights in a reference case study, GEO, which reports competitive but setting-dependent

performance across representative benchmarks and serves as an illustrative example of bridging classical principles with graph-based models.

Through this synthesis, the survey clarifies the current landscape of OOD detection and outlines directions for future research. In particular, it emphasizes the continued integration of classical statistical reasoning with graph-based representation learning as a pathway for developing more reliable OOD detection methods across diverse application domains.

The remainder of this paper is organized as follows. Section II provides background information and introduces benchmark tasks for OOD detection. Section III presents a detailed classification of OOD detection methods into traditional and DL approaches, reviewing representative techniques within each category and presenting a taxonomy summarizing their relationships. Section IV builds on this classification by examining how classical OOD principles are reinterpreted within GNN frameworks, discussing representative GNN-based models and identifying challenges and future directions. Section V describes the illustrative case study, GEO, along with representative experimental results. Finally, Section VI concludes the paper.

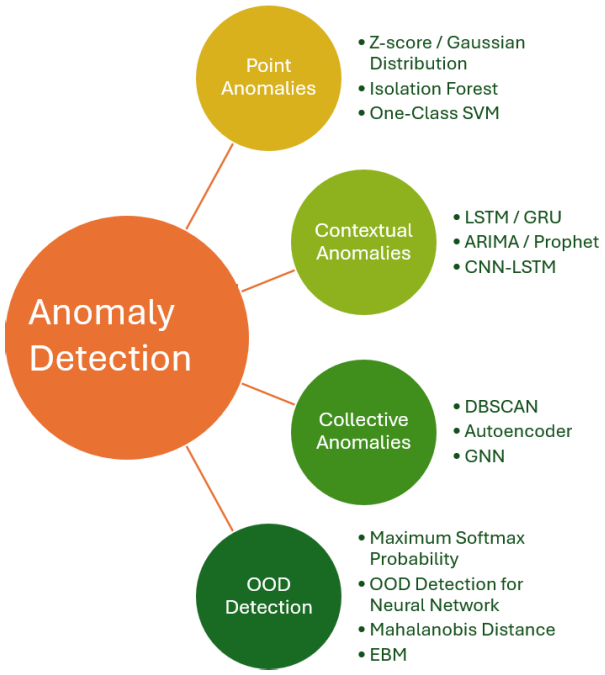

FIGURE 1. Overview of AD and its OOD-detection subset, illustrating typical categories, applications, and methodologies [18].

II. BACKGROUND

This section provides foundational knowledge for OOD detection. We first present an overview of AD and its subcategories, then clarify the role of OOD detection within AD and formalize the problem setting. We next outline its historical evolution from classical to DL and graph-based paradigms. Finally, we summarize representative application domains, benchmark datasets, and commonly used evaluation metrics.

A. OVERVIEW OF AD

Figure 1 illustrates the conceptual hierarchy of AD, which can be broadly categorized into four subgroups: *point*, *contextual*, *collective*, and *OOD* anomalies. This taxonomy reflects how anomalies manifest under diverse data characteristics and real-world contexts. Depending on their manifestation and context, anomalies can take several distinct forms, as summarized below.

1) POINT ANOMALIES

A point anomaly occurs when a single data instance exhibits attributes that significantly deviate from the majority of the dataset. For example, if a user's typical daily credit card spending is about $100, but a transaction of $400 occurs, that instance would be considered anomalous. Common detection methods include *Gaussian* or *Z-score–based screening* [19], [20], [21], ensemble approaches such as *Isolation Forest* [22], and kernel-based techniques such as *One-Class Support Vector Machine* (OCSVM) [23].

2) CONTEXTUAL ANOMALIES

A contextual anomaly refers to an instance that may appear normal in isolation but becomes anomalous when evaluated within a specific context (e.g., time, location, or surrounding conditions). For instance, a temperature reading of 30°C may be normal in summer but abnormal in winter. Typical detection approaches include time-series models such as *AutoRegressive Integrated Moving Average* [24], [25] and *Prophet* [26], recurrent architectures like *Gated Recurrent Unit* (GRU) [27], [28] and *Long Short-Term Memory* (LSTM) [29], [30], as well as hybrid *Convolutional Neural Network* (CNN)–LSTM models [31].

3) COLLECTIVE ANOMALIES

Collective anomalies arise when a group of data points collectively deviates from expected behavior, even though individual points may not appear abnormal on their own. A typical example includes coordinated fake social media accounts or clusters of malicious network activities. Popular detection methods include *Density-Based Spatial Clustering of Applications with Noise* [32], *Autoencoder* (AE) models [33], [34], [35], and GNNs [36], which capture interdependencies among related data instances.

4) OOD DETECTION

OOD detection focuses on identifying inputs drawn from distributions different from the training data, which is essential for assessing model reliability and robustness. Representative approaches include *Maximum Softmax Probability* (MSP) [37], [38], baseline *OOD detectors for neural networks* [39], *Mahalanobis distance–based scoring* [40], and *Energy-Based Models* (EBMs) that assign low energy to *In-Distribution* (ID) samples and higher energy to OOD samples [41]. Classical OOD detection methods typically

assume that data points are independent and identically distributed. However, many real-world systems are inherently relational; examples include fraud rings in transaction networks, botnets in social networks, and abnormal devices in communication graphs. This has motivated graph-aware extensions in which node embeddings are learned via GNNs and OOD scores are computed using adaptations of classical objectives such as one-class classification, density/likelihood estimation, or energy-based scoring on graphs [13], [42], [43], [44], [45].

Among the four anomaly types above, OOD detection is the central focus of this survey. In the remainder of the paper, we trace its evolution from statistical and distance-based methods, through deep generative and energy-based formulations, to recent GNN-based paradigms that explicitly reason over graph structure and interdependent data representations.

B. THE ROLE OF OOD DETECTION WITHIN AD

Having situated OOD detection within the broader AD taxonomy, we treat it in this survey as a distinct reliability objective rather than merely another anomaly subtype. Standard AD often identifies deviations within a known operating distribution, whereas OOD detection asks whether an input comes from a distribution for which the model's learned decision rule should not be trusted [4]. This distinction is central to trustworthy AI systems because overconfident predictions on unseen traffic signs, rare clinical cases, novel attacks, or shifted graph structures can produce unsafe downstream decisions [6].

This reliability view also motivates the graph-aware emphasis of the survey. Traditional methods, typically designed for independent or low-dimensional data, can encounter difficulties when applied to complex, high-dimensional, and interdependent settings [5]. GNNs provide a natural way to incorporate both local and global relational context, but they also require classical OOD principles to be reformulated around graph structure rather than transferred mechanically. In the following subsections, we formalize the OOD detection problem, trace its historical evolution, and summarize representative benchmarks and evaluation metrics.

C. PROBLEM FORMULATION

Machine learning models are typically trained on datasets that cannot fully encompass the complexity of real-world inputs. When these models encounter inputs that differ from their training distribution, they may produce unreliable or overconfident outputs. OOD detection therefore formalizes a reliability-oriented decision problem: recognizing when a model is processing data outside its training distribution rather than merely identifying atypical samples within the same distribution.

Formally, given a classifier g_θ (e.g., a CNN) trained on ID data $\mathcal{D}_{\text{in}}$, the goal of OOD detection is to construct an OOD score $s(\mathbf{x}; g_\theta)$ and a threshold τ that distinguish samples from $\mathcal{D}_{\text{in}}$ from samples drawn from an unknown distribution $\mathcal{D}_{\text{out}}$. Throughout this paper, unless stated otherwise, larger scores indicate stronger OOD evidence. For native quantities with the opposite direction, such as likelihood or confidence, this convention can be obtained by negating the native score or by explicitly stating the threshold direction. The induced binary decision function is expressed in Equation (1).

$$F(\mathbf{x}; g_\theta, \tau) = \begin{cases} 0 & \text{if} \quad s(\mathbf{x}; g_\theta) < \tau \\ 1 & \text{if} \quad s(\mathbf{x}; g_\theta) \geq \tau \end{cases} \tag{1}$$

The history of OOD detection is closely tied to the development of machine learning algorithms and data preprocessing techniques [46]. In earlier studies, which we subsequently discuss, OOD samples were often identified based on simple assumptions, such as a single Gaussian or a mixture of Gaussian distributions. However, in reality, data distributions often do not follow a Gaussian distribution and exhibit much greater complexity.

A more modern approach has generalized the AD problem and treated OOD detection as a subtask based on concepts such as *covariate shift* and *semantic shift* [4]. Covariate shift refers to differences in the observable input space while the labeling function remains unchanged, such as adversarial examples, domain shifts, or style changes. This difference is often used to evaluate model generalization and stability, as test points become more diverse and misleading. In contrast, semantic shift refers to fundamental semantic differences in class attributes or clusters that were not present in the training data. From this perspective, many OOD settings can be viewed as semantic shifts for which a model should abstain, defer, or flag the input rather than return an overconfident prediction.

D. FROM CLASSICAL METHODS TO DL AND GNN-BASED APPROACHES

The formulation above exposes a common structure across OOD detectors: a representation, a scoring rule, and a thresholding mechanism. Classical methods instantiate this structure through density estimates, distances, margins, or reconstruction errors. These tools are often effective in low-dimensional, well-structured settings, but they can become brittle when the relevant OOD evidence is distributed across high-dimensional features or relational dependencies.

DL methods extend this pipeline by learning representations on which OOD scores can be computed. Auxiliary classifiers [39], softmax-based confidence scores [37], AEs [47], GANs [48], ensembles [49], and energy-based scoring [41] all preserve this score-based view while replacing handcrafted features with learned ones. GNN-based methods further adapt the same idea to relational data by allowing the representation and the score to depend on graph structure.

This progression is visible across application settings. In fraud detection, classical Gaussian models estimate

TABLE 1. Representative OOD application domains and common evaluation use cases.

Domain	Typical OOD risk	Common evaluation use case
Autonomous driving and robotics	Unseen objects, unusual road signs, rare weather, sensor artifacts, or unexpected scenes.	Detecting unsafe inputs before perception or planning modules make overconfident predictions.
Medical diagnosis and healthcare	Unseen diseases, acquisition changes, scanner/hospital shifts, and patient population shifts.	Flagging open-set clinical cases or domain-shifted medical images for human review.
Cybersecurity, fraud, and banking	Novel attacks, rare intrusion patterns, abnormal transactions, and coordinated fraud rings.	Treating emerging attack or fraud categories as OOD/anomalous events.
Natural language processing	Out-of-scope intents, domain-shifted corpora, semantic shift, and sentence-pair generalization failures.	Detecting inputs outside the training task or deployment domain.
Industrial and scientific systems	Sensor drift, manufacturing faults, unseen material states, and distribution shifts in scientific measurements.	Monitoring quality-control, predictive-maintenance, and discovery pipelines under novel operating conditions.
Graph-structured applications	OOD nodes, edges, subgraphs, graph labels, topology changes, temporal shifts, and environment shifts.	Evaluating graph OOD detection under feature, structural, semantic, and temporal distribution changes.

transaction likelihoods probabilistically [50], [51], whereas GNNs can additionally capture relational patterns among accounts. In network intrusion detection, EBMs detect deviations in network states [41], [52], while graph-based models can incorporate the underlying network topology. In medical diagnosis, architectures such as ChebNet and graph convolutional networks analyze structured data including medical records and molecular structures [42], [53]. These examples motivate the method-level taxonomy developed in the next section.

E. APPLICATIONS, BENCHMARKS, AND METRICS

OOD detection plays a critical role in real-world applications where model reliability and robustness are essential. Table 1 summarizes representative domains and the types of OOD risk they are commonly used to evaluate.

To evaluate OOD detection models, researchers use benchmark datasets containing both ID and OOD samples. Tables 2 and 3 separate commonly used non-graph and graph benchmark statistics so that dataset scale is explicit without mixing it with application-level use cases.

In practice, non-graph OOD benchmarks often pair ID classification datasets with semantically distant or near-OOD test sets, whereas graph OOD benchmarks commonly evaluate label, feature, structural, temporal, covariate, concept, or environment shifts. †For graph datasets, edge counts may differ across software libraries because some implementations store reverse edges explicitly. Unless otherwise stated, Table 3 reports dataset-source or common undirected/reference counts for citation, co-purchase, and coauthor datasets, while `ogbn-arxiv` is reported as a directed *Open Graph Benchmark* (OGB) citation graph.

Standard evaluation metrics for OOD detection include the *Area Under the Receiver Operating Characteristic Curve* (AUROC), *Area Under the Precision-Recall Curve* (AUPR), *Area Under the Risk-Coverage Curve* (AURC), and F-scores. Another commonly used metric is FPR@TPRx, which reports the *False Positive Rate* (FPR) at a given *True Positive Rate* (TPR) threshold x.

For a binary OOD decision, let OOD samples be the positive class and let $s(\mathbf{x})$ denote an OOD score, where larger values indicate stronger OOD evidence. At a threshold τ, samples with $s(\mathbf{x}) \geq \tau$ are predicted as OOD. The main threshold-dependent quantities are summarized in Equation (2).

$$\begin{aligned}
\mathrm{TPR}(\tau) &= \frac{\mathrm{TP}(\tau)}{\mathrm{TP}(\tau)+\mathrm{FN}(\tau)},\\
\mathrm{FPR}(\tau) &= \frac{\mathrm{FP}(\tau)}{\mathrm{FP}(\tau)+\mathrm{TN}(\tau)},\\
\mathrm{Precision}(\tau) &= \frac{\mathrm{TP}(\tau)}{\mathrm{TP}(\tau)+\mathrm{FP}(\tau)},\\
F_1(\tau) &= \frac{2\,\mathrm{Precision}(\tau)\,\mathrm{TPR}(\tau)}{\mathrm{Precision}(\tau)+\mathrm{TPR}(\tau)}.
\end{aligned} \tag{2}$$

Here, TP, FP, TN, and FN denote true positives, false positives, true negatives, and false negatives, respectively. AUROC and AUPR summarize threshold-swept performance as shown in Equation (3).

$$\begin{aligned}
\mathrm{AUROC} &= \int_0^1 \mathrm{TPR}(u)\,du,\\
\mathrm{AUPR} &= \int_0^1 \mathrm{Precision}(r)\,dr,
\end{aligned} \tag{3}$$

where u denotes the false-positive-rate axis and r denotes recall. FPR@TPRx is computed by selecting a threshold τ_x such that $\mathrm{TPR}(\tau_x) = x$ and then reporting $\mathrm{FPR}(\tau_x)$. For selective prediction, AURC integrates the risk–coverage curve in Equation (4).

$$\mathrm{AURC} = \int_0^1 R(c)\,dc, \tag{4}$$

where c is the retained coverage and $R(c)$ is the prediction risk at that coverage.

With these definitions, benchmarks, and metrics in place, the following section classifies OOD detection methods into distinct methodological families and highlights how their assumptions, scoring rules, and limitations differ.

III. CLASSIFICATION OF OOD DETECTION METHODS

This section provides the methodological foundation for the survey. We organize OOD detection methods into two broad families: *traditional* methods and DL methods. For each family, we describe representative techniques, their mathematical formulations, and their respective strengths and limitations.

TABLE 2. Representative non-graph OOD benchmarks and reported dataset statistics.

Domain	Dataset	Scale	Split/secondary	Labels	Input/statistic type
Computer vision	CIFAR-10 [54]	50,000	10,000 test	10	32×32 RGB images
	CIFAR-100 [54]	50,000	10,000 test	100	32×32 RGB images
	MNIST [55]	60,000	10,000 test	10	28×28 grayscale images
	Fashion-MNIST [56]	60,000	10,000 test	10	28×28 grayscale images
	ImageNet/ILSVRC [57]	1,281,167	50,000 val.; 100,000 test	1,000	Natural images
Language	GLUE [58]	9 tasks	Task-dependent	Task-dependent	Sentence-pair language tasks
	SNLI [59]	550,152	10,000 val.; 10,000 test	3	Sentence pairs
Medical imaging	CheXpert [60]	224,316	65,240 patients	14	Chest radiographs
	ChestX-ray14 [61]	112,120	30,805 patients	14	Frontal chest X-rays
	MedMNIST v2 [62]	∼708K 2D	∼10K 3D	Dataset-dependent	12 2D and 6 3D datasets
Cybersecurity/AD	KDDCup99 [63]	4,898,431	311,027 test	Attack/normal	41 traffic features
	NSL-KDD [64]	125,973	22,544 test	5	41 traffic features

TABLE 3. Representative graph OOD benchmark families and reported dataset statistics.

Benchmark family	Nodes	Edges†	Features	Classes	Dataset source/type
Cora [65]	2,708	5,429	1,433	7	Citation network
CiteSeer [65]	3,327	4,732	3,703	6	Citation network
PubMed [65]	19,717	44,338	500	3	Citation network
Amazon-Photo [66]	7,650	119,081	745	8	Co-purchase graph
Coauthor-CS [66]	18,333	81,894	6,805	15	Co-authorship graph
`ogbn-arxiv` [67]	169,343	1,166,243	128	40	Directed citation graph
GOOD [68]	11 datasets; 17 domain selections; 51 splits				Standardized graph OOD benchmark suite

This organization identifies the classical scores, objectives, and assumptions that will later be reinterpreted in graph-aware settings. A comprehensive taxonomy summarizing this classification is presented at the end of the section.

A. TRADITIONAL METHODS

Traditional OOD methods generally operate on explicit statistical, geometric, or margin-based assumptions about the ID data. We group them into density/probabilistic models, distance-based models, and one-class classification methods. This order moves from global distributional modeling, to local neighborhood comparison, and finally to direct estimation of the normal data support.

1) CLASSICAL DENSITY ESTIMATION AND PROBABILISTIC MODELS

Density estimation and probabilistic models play a fundamental role in anomaly and OOD detection by modeling the underlying data distribution and identifying instances that significantly deviate from it. Representative approaches in this category include the classical Gaussian model, *Gaussian Mixture Models* (GMMs), and *Kernel Density Estimation* (KDE). These methods are valued for their simplicity, interpretability, and solid theoretical foundations. In low-dimensional, well-behaved data settings, they can provide efficient and reliable OOD detection. However, their effectiveness is fundamentally limited by several factors: (i) strong distributional assumptions in some models (e.g., Gaussianity) [69], (ii) poor scalability and performance on high-dimensional or complex data due to the curse of dimensionality [70], and (iii) limited flexibility in modeling multimodal or structured data [71], [72]. Overall, density estimation and probabilistic models remain important classical baselines, but their limitations in real-world, large-scale applications have motivated the development of more advanced data-driven approaches.

a: CLASSIC GAUSSIAN MODEL

One of the fundamental methods for multivariate AD is calculating the Mahalanobis distance [73] from a test point to the mean of the training data [74]. It is a measure of the distance between a point and a distribution defined by Equation (5).

$$D_M(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})} \tag{5}$$

This is equivalent to modeling a multivariate Gaussian distribution for the training data and assessing the probability of occurrence for an input data point based on that model. The technique using a multivariate Gaussian distribution [50] is one of the most basic approaches to determine whether inputs are OOD. The Gaussian density is defined in Equation (6) [51].

$$p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \times \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right) \tag{6}$$

Here, $\mathbf{x} \in \mathbb{R}^d$ is the input vector, $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean vector, $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ is the covariance matrix, $|\boldsymbol{\Sigma}|$ is its

determinant, and $\boldsymbol{\Sigma}^{-1}$ is its inverse. This compact notation is sufficient for the rest of the survey because subsequent methods reuse the same density, distance, and score conventions. This method uses the *Maximum Likelihood Estimation* (MLE) [75] technique to estimate the mean and covariance of a given dataset representing the ID data. To identify whether a new data point is anomalous, a threshold can be applied to the likelihood or negative log-likelihood of the sample, depending on the chosen score direction. For a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, the likelihood is computed as Equation (7).

$$L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{n} p(\mathbf{x}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{7}$$

Applying the natural logarithm to the density function p yields the log-likelihood in Equation (8).

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{nd}{2}\ln(2\pi) - \frac{n}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) \tag{8}$$

By taking derivatives and setting them to zero, the maximum likelihood estimates for the parameters are obtained as given in Equation (9).

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i, \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top. \tag{9}$$

The multivariate Gaussian distribution is widely used in statistics and machine learning [50]. The central limit theorem [76] also shows why Gaussian assumptions can be reasonable in some settings, as sums of independent random variables tend toward a Gaussian distribution. This analytical tractability makes Gaussian modeling a useful baseline, with likelihood computation typically cheaper than more complex density estimators [77].

Later, Lee et al. [78] showed that pre-trained features generated by *Deep Neural Networks* can be fitted well to a Gaussian distribution and proposed computing a confidence score using the Mahalanobis distance to detect anomalies. While their approach learns the distribution parameters only after training the classifier, the *Gaussian Likelihood Out of Distribution* [79] technique estimates the parameters of the Gaussian model during the training phase. Subsequently, the Gaussian likelihood and the log-likelihood ratio are employed to detect OOD samples. While a multivariate Gaussian model can serve as a simple baseline for OOD detection, this model is only applicable to data that satisfy the Gaussian assumptions. **Strengths:** Simple, efficient for low-dimensional data, and easy to implement. **Weaknesses:** Assumes data are Gaussian-distributed; performs poorly for multimodal or high-dimensional data; and is sensitive to outliers and assumption violations [69], [71], [72].

b: MIXTURE OF GAUSSIAN MODEL

Although the basic Gaussian distribution has useful analytical properties, a single Gaussian often cannot capture complex or multimodal datasets observed in real-world scenarios. GMMs [80], [81] address this limitation by representing a distribution as a weighted combination of multiple Gaussian components. Data points that do not align well with any component [82] may be considered anomalous, as expressed by Equation (10).

$$p(\mathbf{x}; \lambda) = \sum_{i=1}^{M} w_i g(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{10}$$

where $\mathbf{x}$ is a d-dimensional continuous-value vector, w_i are the mixture weights, and $g(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, $i = 1, \ldots, M$, are the component Gaussian densities. Each component density is a d-variate Gaussian function of Equation (11).

$$g(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \times \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)\right) \tag{11}$$

The approach to detect anomalies in this method is similar to that applied in the classic Gaussian model. GMMs estimate three fundamental parameters for each Gaussian component: the mean vectors, which represent the central location of the data; the covariance matrices, which capture the spread, shape, and potential correlations between features; and the mixing coefficients, which indicate the proportion of the overall data distribution accounted for by each component. These parameters are commonly estimated using the iterative *Expectation-Maximization* algorithm [83], [84] or MLE [75]. A test sample is considered anomalous if its log-likelihood value is below the specified threshold. Although GMMs are more flexible than a single Gaussian model, each component still relies on Gaussian assumptions.

Several studies combine GMMs with representation-learning techniques to improve detection performance, such as the *Deep Autoencoding Gaussian Mixture Model* [85]. This model extracts key low-dimensional features with an AE and feeds them to the GMM to compute the input likelihood. In summary, while GMMs offer greater flexibility than the classic Gaussian model, they still exhibit notable limitations. **Strengths:** Flexible for multimodal distributions; widely used in practice; capable of modeling data structures beyond a single Gaussian. **Weaknesses:** Still assumes Gaussianity within each component; sensitive to initialization and the curse of dimensionality; requires careful tuning of the number of components; computationally intensive for large datasets [71].

c: KERNEL DENSITY ESTIMATION

KDE is a non-parametric method used in statistics for probability density estimation. It applies kernel smoothing to estimate the probability density function of a random variable based on kernels as weights. This method allows inferences about the population to be made based on a finite data sample. KDE is also known as the Parzen–Rosenblatt window method, named after Emanuel Parzen and Murray Rosenblatt,

who are credited with independently creating it in its current form [86]. A well-known use of KDE is to calculate the class-conditional marginal densities of data in the context of a naive Bayes classifier [87], thereby enhancing the precision of its predictions [88]. Unlike the classic Gaussian model and GMMs, KDE does not require the input data to satisfy a Gaussian assumption. The OOD decision rule is still density-based, but the *Probability Density Function* (PDF) of a dataset $X = \{x_1, x_2, \ldots, x_n\}$ is estimated non-parametrically. The density value at a query point x using KDE is defined by Equation (12).

$$\widehat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{12}$$

where K is the kernel, a non-negative function, and $h > 0$ is a smoothing parameter called the bandwidth.

A kernel with subscript h is called the scaled kernel and defined as $K_h(x) = \frac{1}{h}K\left(\frac{x}{h}\right)$. Intuitively, K is a kernel function satisfying non-negativity and normalization conditions [89]. A wide range of kernel functions can be used in KDE, with the most common being the Gaussian and Epanechnikov kernels [90]. Additionally, the bandwidth h controls the smoothness of the shape density function; the greater the value h, the smoother the shape of the density function f. A new instance, which belongs to the low probability region of this PDF, is declared to be anomalous.

Some studies calculate the PDF of activation at different network layers by applying KDE on the data from the same distribution. At test time, they assign a confidence score for each layer and combine them with logistic regression to get the final score [91]. Another study uses variational autoencoding and KDE to deal with high-dimensional data. This study extracts low-dimensional features from the input data and then applies KDE to estimate the PDF of the training data [92]. Kernel-based methods have at least quadratic complexity in the number of samples [93] unless they use some approximation technique [94]. In summary, while KDE offers distribution-free flexibility, it also faces notable computational and scalability challenges: **Strengths:** Non-parametric; can model arbitrary data distributions; does not assume any specific distributional form. **Weaknesses:** Extremely sensitive to the bandwidth parameter; suffers from the curse of dimensionality (ineffective for high-dimensional data); high time complexity without approximation methods; computationally expensive for large datasets [70], [71].

2) DISTANCE-BASED MODELS

The second group of traditional methods are distance-based models. Generally, distance-based models are a class of OOD detection approaches that assess the abnormality of data points by measuring their distances to neighboring points or regions in the feature space. Two prominent techniques within this category are the *Local Outlier Factor* (LOF) algorithm and the Isolation Forest method. Distance-based models offer intuitive approaches to AD by leveraging distance metrics to identify data points that deviate significantly from their local neighborhoods or expected patterns. They are valued for their flexibility and distribution-free nature, but often struggle with computational cost and reduced reliability in high-dimensional data spaces.

a: LOCAL OUTLIER FACTOR

The LOF algorithm is a type of unsupervised AD method. It assesses the deviation in local density of a specific data point in comparison to its neighboring points. It identifies samples as outliers if their density significantly falls below that of their neighboring points. This non-parametric technique measures the degree of outlierness of each data point based on its local density compared to its neighbors [95]. The LOF of a point is defined as the ratio between the average local reachability density of its k nearest neighbors and its own local reachability density. Let $N_k(x)$ denote the k nearest neighbors of x. The formula is described in Equation (13) [96].

$$\mathrm{LOF}_k(x) = \frac{\frac{1}{|N_k(x)|}\sum_{y\in N_k(x)} \mathrm{lrd}_k(y)}{\mathrm{lrd}_k(x)} \tag{13}$$

The local reachability density is expressed by Equation (14).

$$\mathrm{lrd}_k(x) = \left(\frac{1}{|N_k(x)|}\sum_{y\in N_k(x)} \mathrm{rd}_k(x, y)\right)^{-1} \tag{14}$$

The reachability distance between two points is defined in Equation (15).

$$\mathrm{rd}_k(x, y) = \max\{d_k(y), d(x, y)\} \tag{15}$$

where $d(x, y)$ is the distance between x and y, and $d_k(y)$ is the distance from y to its k-th nearest neighbor. A high LOF value indicates that x has lower local density than its neighbors and is therefore more likely to be an outlier.

The LOF algorithm has several advantages, including its ability to handle data with varying densities and shapes and its relative robustness to noise and irrelevant attributes. It has been applied effectively in settings such as network intrusion detection [97] and processed classification benchmark data [98]. However, it also has drawbacks, such as requiring a user-defined parameter k, which can affect performance and interpretation, and the fact that the threshold at which a LOF value greater than 1 indicates an outlier is not fixed across datasets. Several extensions of LOF address these limitations; for example, Schubert et al. [98] abstract various local outlier detection methods into a general framework applicable to spatial, video, and network outlier detection. In summary, LOF is effective in detecting local anomalies, particularly in datasets with heterogeneous densities or structures [71]. However, its performance is heavily influenced by the choice of hyperparameters and suffers in high-dimensional or large-scale data due to increased computational demands and diminished effectiveness of distance metrics.

b: ISOLATION FOREST

Approaches mentioned above generally construct a pattern of normal instances, then deem a data point anomalous if it deviates from the normal profile [80], [81], [90], [99]. In contrast, the Isolation Forest method isolates and identifies anomalies based on the idea that anomalies are "*few-and-different*" [100].

Isolation Forest-based AD is a two-phase procedure. In the first phase, the method builds an ensemble of isolation trees. The partitions in each tree are generated by randomly selecting a feature and then choosing a split value between the minimum and maximum values of the selected feature. This differs from the partitioning strategies used in decision trees and random forests, which select features/splits based on impurity reduction (e.g., Gini impurity) at each node [101], [102]. In the second phase, given a sample of data $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ of n instances from a d-variate distribution, the Isolation Forest anomaly score s_{IF} of an instance $\mathbf{x}$ is defined by Equation (16).

$$s_{\text{IF}}(\mathbf{x}, n) = 2^{-\frac{\mathbb{E}[h(\mathbf{x})]}{c(n)}} \tag{16}$$

where $\mathbb{E}[h(\mathbf{x})]$ is the average path length of $\mathbf{x}$ over a forest of isolation trees, and the normalization constant $c(n)$ is given in Equation (17).

$$c(n) = 2\,H_{n-1} - \frac{2(n-1)}{n} \tag{17}$$

is the average path length of an unsuccessful search in a binary search tree with n instances (with $H_m = \sum_{i=1}^{m} \frac{1}{i}$) [100]. Points that travel deeper into the trees are more likely to be normal and thus receive smaller anomaly scores. The score lies in $[0, 1]$. As suggested by Liu et al. [100], scores close to 1 typically indicate anomalies, whereas scores well below 0.5 indicate normal instances; scores around 0.5 suggest no clear anomalies.

More recently, there have been many extensions of Isolation Forest that attempt to obtain better detection performance by employing hierarchical criteria for the selection of dividing thresholds or dimensions [103], [104], [105], using optimal random hyperplanes [106], [107], [108], or leveraging neural networks to map original data to new representation spaces, such as *Deep Isolation Forest* [22]. Moreover, to deal with imprecise, incomplete, ambiguous, or uncertain data, fuzzy variants have been proposed [109].

In summary, Isolation Forest is widely adopted due to its efficiency and scalability for large datasets. **Strengths:** Fast and scalable; effective on large datasets; few hyperparameters. **Weaknesses:** Less effective for subtle/local outliers; performance degrades in high-dimensional settings or with many irrelevant features (random splits become less informative) [71], [72].

3) ONE-CLASS CLASSIFICATION

One-class classification models the *support* of the ID distribution without requiring explicit negatives. Unlike density/probabilistic estimators in Section III-A1, which target $p(x)$, and distance-based methods in Section III-A1c, which rely on local geometry, one-class approaches learn a compact normal region and judge samples by their conformity to it. This setting is natural when anomalies are rare or ill-specified and benefits strongly from good representations. We next cover two representative lines: (a) *OCSVM*, which separates normal data from the origin via a maximum-margin principle, and (b) *Support Vector Data Description* (SVDD)-style methods, including Deep SVDD, that concentrate learned features within a small-volume hypersphere.

a: ONE-CLASS SUPPORT VECTOR MACHINE

In one-class classification [110], a model is trained with a well-characterized positive class and a poorly represented or sparsely populated negative class. OCSVM [46], [111], [112] is an extension of traditional *Support Vector Machines* (SVMs) designed for AD, where the majority of the data belongs to one class (normal) and only a small fraction belongs to the other class (anomaly). The objective of the OCSVM is to find the hyperplane that best separates the normal data from the origin while maximizing the margin. The decision function is defined in Equation (18).

$$f(\mathbf{x}) = \text{sign}\left(\mathbf{w}^\top \phi(\mathbf{x}) - \rho\right) \tag{18}$$

where $\mathbf{w}$ is the normal vector to the hyperplane, $\phi(\mathbf{x})$ is the (possibly implicit) feature mapping induced by the kernel, and ρ denotes the offset (distance from the origin) defining the decision boundary. The optimization problem is formulated [112] in Equation (19).

$$\begin{aligned} &\underset{\mathbf{w},\,\xi,\,\rho}{\text{minimize}}\ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i - \rho \\ &\text{subject to } \mathbf{w}^\top \phi(\mathbf{x}_i) \geq \rho - \xi_i, \quad i = 1, \ldots, m, \\ &\xi_i \geq 0 \end{aligned} \tag{19}$$

Here, $\nu \in (0, 1]$ provides an upper bound on the fraction of outliers and a lower bound on the fraction of support vectors. Tax and Duin [113] proposed a method that uses artificially generated outliers to adjust OCSVM parameters, aiming to avoid both over-fitting and under-fitting. Yu [114] presented a one-class classification algorithm that uses SVMs with positive and unlabeled data, eliminating the need for labeled negative data. Manevitz and Yousef [115] highlighted limitations of OCSVM-based one-class classification, including the need for sufficient training data to learn a reliable class boundary without labeled negatives. Li et al. [116] improved the OCSVM [70] approach for AD in intrusion detection systems, achieving higher accuracy by considering all points close to the origin as outliers. Hao [117] combined fuzzy set theory with OCSVM [70] to address the sensitivity of standard OCSVM to outliers and noise.

b: DEEP ONE-CLASS CLASSIFICATION

Classical AD methods such as OCSVM [70] or KDE [99] often fail in high-dimensional, data-rich scenarios because

of poor computational scalability and the curse of dimensionality. Deep one-class classification methods aim to learn effective representations for AD in an unsupervised manner. Many deep AD approaches rely on reconstruction-error heuristics from AEs or GANs, but these heuristics do not directly optimize a compact normality objective.

In contrast, Ruff et al. [118] introduced Deep SVDD, a deep one-class objective that maps normal examples toward a compact hypersphere with minimal volume. By minimizing the hypersphere enclosing the learned representations of the training data, the network is encouraged to identify common factors of the normal data distribution. The soft-boundary Deep SVDD objective function is defined in Equation (20).

$$\min_{R,W} R^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\max\{0, \|\phi(x_i; W) - c\|_2^2 - R^2\} + \frac{\lambda}{2}\sum_{\ell=1}^{L}\|W_\ell\|_F^2 \tag{20}$$

The network feature mapping of input x_i, denoted by $\phi(x_i; W)$, is obtained by applying the neural network parameters W to the input. The neural network has L layers and its parameters are regularized by the weight decay term $\lambda > 0$. The network feature mapping is constrained to lie on a hypersphere with radius R and center c, where R is determined by the parameter $\nu \in (0, 1]$. The parameter ν balances the trade-off between minimizing the sphere volume and minimizing the boundary violations. We use $\|\cdot\|_2$ for the Euclidean norm and $\|\cdot\|_F$ for the Frobenius norm.

B. DL METHODS

DL methods retain the same broad OOD objective but replace fixed feature spaces with learned representations. This shift allows reconstruction errors, likelihoods, gradient signals, energy scores, and graph-based embeddings to be computed in spaces shaped by neural training. The following subsections review reconstruction models, generative models, gradient-based methods, and GNN backbones, setting up the graph-aware synthesis in Section IV.

1) RECONSTRUCTION MODELS

Reconstruction-based AD methods rely on the premise that models trained on ID data struggle to effectively reconstruct OOD samples. This divergence in model performance between ID and OOD data serves as a valuable indicator for AD. Two prominent techniques in this category are AEs and *Principal Component Analysis* (PCA).

a: AUTOENCODER

The fundamental concept behind reconstruction-based methodologies lies in the observation that the encoder-decoder framework [119], when trained on ID data, typically produces disparate outcomes for ID and OOD samples. This divergence in model performance serves as a valuable indicator for AD. Specifically, models exclusively trained on ID data struggle to effectively reconstruct OOD data [40], facilitating the identification of OOD instances. While pixel-level comparison in reconstruction models is not widely favored in OOD detection due to its high training cost, an alternative approach involving the reconstruction of hidden features has been proposed as a promising solution [120]. Recent research, exemplified by MoodCat [121], takes a novel approach by masking a random portion of the input image and discerning OOD samples based on the quality of classification-driven reconstruction results. Because direct kernel density estimation can be ineffective in high-dimensional spaces [40], [86], [122], [123], another line of work creates compact latent representations [124], [125] and minimizes regularization loss to ensure that latent features of the ID are distributed within a specific space. Let $\boldsymbol{x}$ be an input to an AE model, and let $\mathrm{Enc}(\boldsymbol{x})$ denote the encoded representation of $\boldsymbol{x}$. The reconstruction error of the AE is formulated as a function $R(\boldsymbol{x})$, given by Equation (21).

$$R(\boldsymbol{x}) = \mathrm{Dist}\big(\boldsymbol{x}, \mathrm{Dec}(\mathrm{Enc}(\boldsymbol{x}))\big) \tag{21}$$

where $\mathrm{Dec}(\cdot)$ is the decoding function of the AE, and $\mathrm{Dist}(\cdot,\cdot)$ is an appropriate distance metric used to compute the reconstruction error.

b: PRINCIPAL COMPONENT ANALYSIS

PCA [126] is widely used for dimensionality reduction and feature extraction. This technique finds an orthonormal basis in which the dominant variation in the data is captured by a small number of principal components, while less informative directions are suppressed [127], [128]. The original data are projected onto these principal components by maximizing projected variance, as formulated in Equation (22) [51].

$$\max_{W\in\mathbb{R}^{D\times d}} \mathrm{tr}(W^\top C W) \quad \text{s.t.} \quad W^\top W = I_d \tag{22}$$

Here $W = [w_1, \ldots, w_d]$ is an orthonormal basis for the d-dimensional principal subspace in data space $\mathcal{X} \subseteq \mathbb{R}^D$, $\bar{x}$ is the sample mean, and C is the covariance matrix of $X = \{x_1, \ldots, x_n\}$. The standard approach to solve this optimization problem is to use the Lagrange multiplier method [129], [130], yielding the eigenvalue equation in Equation (23).

$$Cw = \lambda w \tag{23}$$

The equation means that each principal direction w must be an eigenvector of C, and λ is its corresponding eigenvalue [130]. The $d \leq D$ eigenvectors with the largest eigenvalues determine the principal subspace spanned by W.

The *Squared Prediction Error* (SPE) [131], [132] is computed for each data point using Equation (24), measuring the difference between the original data point and its projection onto the principal subspace spanned by W. OOD data tends to yield higher reconstruction error than ID data.

$$\mathrm{SPE}(x) = \|x - WW^\top x\|_2^2 \tag{24}$$

Kernel PCA [133] was developed as an extension of PCA to cover non-linear structures in data by leveraging

the "kernel trick". The reconstruction error, which can be determined via the dual [134], serves as the anomaly score in Kernel PCA [128]. Furthermore, to address the issues of data contamination and noise, several Robust PCA variants have been proposed as well [135], [136]. In addition, the research [137] introduces an extension of PCA called the *Karhunen–Loeve* expansion [138], which takes into account temporal correlations. The Karhunen–Loeve expansion with a Galerkin method is applied to network traffic data. Compared to basic PCA, it gives improved AD performance when the temporal correlation of traffic data is considered.

Recent reconstruction-oriented methods for hyperspectral AD have moved beyond generic autoencoder-style formulations toward self-supervised background modeling. For example, NL2Net employs a dual-branch blind-spot architecture that couples local feature extraction with nonlocal dependency modeling to reconstruct background structure more faithfully and improve anomaly separability in hyperspectral images [139]. This line of work suggests that reconstruction-oriented AD is increasingly incorporating richer feature coupling and background-aware reconstruction objectives.

2) GENERATIVE MODELS

The second notable branch of DL approaches is generative models. In AD and OOD detection, generative models aim to learn the regular data distribution and identify samples that deviate from it. These models provide flexible architectures for modeling complex data patterns and assigning anomaly or OOD evidence through likelihoods, reconstruction behavior, discriminator responses, or energy scores. There are four primary types of such models: EBMs, *Variational Autoencoders* (VAEs), GANs, and *Normalizing Flows* (NFs).

a: ENERGY-BASED MODELS

The first deep EBM networks, such as *Deep Belief Networks* [140] and *Deep Boltzmann Machines* [141], are graphical models consisting of layers of hidden states followed by an output layer to model observed data. Here, the energy function depends not only on the input x but also on the hidden state z, so the energy function takes the form $\mathcal{E}_\theta(x, z)$. The core idea of EBMs [142] is to associate each input with an energy value, where lower energy is assigned to more plausible or ID samples and higher energy is assigned to less plausible or OOD samples. These models can be trained by approximating the gradient of the log-likelihood $\nabla_\theta \log p_\theta(x)$ using techniques such as *Markov Chain Monte Carlo* [143] or score matching [144]. Subsequent studies have replaced probabilistic hidden layers with deterministic hidden layers [145], enabling the use of the energy function $E_\theta(x)$ as an anomaly score [146]. A temperature-scaled conditional distribution can be written in terms of class-conditional energy as shown in Equation (25).

$$p(y \mid \mathbf{x}) = \frac{\exp(-E(\mathbf{x}, y)/T)}{\sum_{y'} \exp(-E(\mathbf{x}, y')/T)} = \frac{\exp(-E(\mathbf{x}, y)/T)}{\exp(-E(\mathbf{x})/T)} \tag{25}$$

The denominator, $\sum_{y'} \exp(-E(\mathbf{x}, y')/T)$, is the partition function over possible labels and involves the temperature parameter T. The corresponding Helmholtz free energy for a data point $\mathbf{x} \in \mathbb{R}^D$ is the negative temperature-scaled logarithm of this partition function, as given in Equation (26).

$$E(\mathbf{x}) = -T \log \sum_{y'} \exp(-E(\mathbf{x}, y')/T). \tag{26}$$

EBMs establish an energy function that allocates low energies to instances within the distribution [142] and higher energies to those outside the distribution. Under the score convention introduced in Section II, the free energy itself can be used as an OOD score: higher values provide stronger OOD evidence. At test time, examples with much higher energies than the training data can therefore be flagged as OOD or anomalous. A threshold can be set on the energy to decide what is OOD. EBM-based OOD detection has demonstrated efficacy across various data types, including images, text, and tabular datasets, often improving over baselines based on softmax confidence or reconstruction errors [39], [41]. Important challenges include selecting reliable energy thresholds and addressing training instability. Noise-based training [147] and ensemble methods [52] have been proposed to improve stability. More recently, EBMs have also been used to reinterpret classification models through energy objectives, which can improve both training behavior and OOD detection performance [41], [148].

b: VAE & GAN

VAE: VAEs are generative models designed for latent variable modeling, incorporating probabilistic principles to facilitate data generation and representation learning. The fundamental idea behind VAEs is to introduce a probabilistic interpretation to the latent variables. Unlike other generative methods, which typically produce a deterministic latent representation [149], [150], VAEs model the latent space as a probability distribution. This allows for greater flexibility in capturing the inherent uncertainty and diversity within the data [151], [152].

The key equation governing the VAE is derived from the principles of variational inference. The objective is to maximize the *Evidence Lower Bound* (ELBO) [153], which is equivalent to minimizing the Kullback-Leibler divergence between the true posterior distribution of the latent variables and an approximating distribution. According to [154], for a given dataset X and latent variable Z, the VAE objective function is expressed in Equation (27).

$$\mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}\right) \simeq \mathbb{E}_{z \sim q_\theta(z|\mathbf{x}^{(i)})}\left[\log p_\phi\left(\mathbf{x}^{(i)} \mid z\right)\right] - D_{\mathrm{KL}}\left(q_\theta\left(z \mid \mathbf{x}^{(i)}\right) \| p(z)\right) \tag{27}$$

The parameters ϕ and θ are optimized during the training process. VAEs can be used for AD by modeling the normal data distribution with the VAE, then looking for test samples

that have high reconstruction error, meaning they deviate significantly from the learned normal distribution [155].

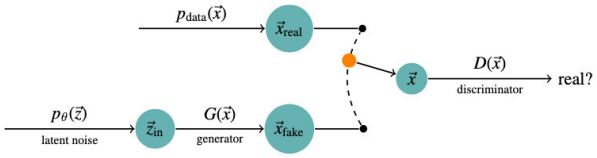

FIGURE 2. The GAN discriminator and generator branches to detect anomalies [48].

GAN: GANs can also support AD by training a discriminator to distinguish real or normal samples from generated samples. Anomalous samples are therefore more likely to receive discriminator responses that differ from the normal data manifold [48]. A GAN consists of a generator G, which maps latent noise $z \sim p_z(z)$ to synthetic samples, and a discriminator D, which distinguishes samples from the data distribution $p_d(x)$ from generated samples. Figure 2 depicts these generator and discriminator branches in the AD setting. The interaction between G and D is commonly formulated as a minimax game [156], [157], where the value function $V(G, D)$ is optimized as shown in Equation (28).

$$\min_G \max_D V(G, D) = \mathbb{E}_{\boldsymbol{x}\sim p_d(\boldsymbol{x})}[\ln D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\ln(1 - D(G(\boldsymbol{z})))] \tag{28}$$

Unlike VAEs, GANs do not directly assign normalized likelihoods to input data points. Applying GANs to AD or OOD detection therefore usually requires intermediate scoring mechanisms, such as discriminator responses or latent-space reconstruction. Another approach is to use GAN-related variants such as *Adversarial Autoencoder* [158] or AnoGAN [34], which estimate anomaly evidence based on the correspondence between an input and its latent representation $z \sim Q$.

GAN-based methods have achieved strong results in AD, especially for image data. Their many variants provide different scoring mechanisms and architectural choices for different data modalities.

Recent GAN-based extensions have also moved beyond direct spectrum-to-spectrum reconstruction toward domain-transformation and mapping-based designs. For instance, *Frequency-to-Spectrum Mapping GAN* is a semisupervised method for hyperspectral AD that maps coarsely selected background spectra to the fractional Fourier domain and trains a GAN to restore them back to the original spectral domain [159]. The key idea is that the model can restore background samples more faithfully than anomaly samples, since the fractional Fourier domain enhances background-anomaly separability and the generator is trained on paired background spectral and frequency-domain samples. During inference, the restored image is subtracted from the original image to form a differential image, and a global Reed–Xiaoli detector is then applied to obtain the final anomaly map. This example shows that GAN-based AD can incorporate semisupervised background modeling and domain-specific transformation objectives rather than relying only on direct self-reconstruction in the original spectral domain.

c: NORMALIZING FLOWS

NFs are a type of deep generative model that can learn complex data distributions. They were introduced in some important papers [160], [161], [162] and explained in their general form [163], which can approximate a target distribution $p^*(x)$ through an invertible transformation f applied to a base distribution $p_Z(z)$ in the latent space. The main idea is using flexible, invertible mappings that can generate data points $x \approx f(z)$ according to the desired distribution $p^*(x)$.

NFs model latent samples as points in a D-dimensional space, matching the dimensionality of the input space. The architecture comprises L invertible layers, denoted as $f_{i,\omega_i} : \mathbb{R}^D \to \mathbb{R}^D$, where $f_\omega = f_{L,\omega_L} \circ \cdots \circ f_{1,\omega_1}$ and each f_{i,ω_i} is invertible for all ω_i. Using the change-of-variables formula, the likelihoods for an input x and a dataset $\mathcal{D}$ are shown in Equation (29).

$$p_X(x) = p_Z\left(f^{-1}(x)\right)\left|\det \frac{\partial f^{-1}}{\partial x}\right|, \quad p(\mathcal{D}) = \prod_{x\in\mathcal{D}} p_X(x) \tag{29}$$

The training objective for the NFs in Equation (29) conventionally involves the maximization of the log-likelihood function, denoted as $\mathcal{L}$, with respect to the parameters governing the invertible transformation f. Specifically, the optimization objective is expressed as given in Equation (30).

$$\underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta) = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{x\sim\mathcal{D}}\left[\log p_X(x;\theta)\right] \tag{30}$$

where θ represents the parameters of the invertible transformation f, and $\mathcal{D}$ denotes the training dataset. The objective seeks to enhance the model's capacity to accurately capture the underlying distribution of the training data by iteratively adjusting the parameters to maximize the log-likelihood.

An advantage of these models, relative to many alternative generative approaches, is their ability to compute exact likelihoods through the change-of-variables formula while retaining efficient sampling properties. Given the exact computability of the probability density function $p_X(x)$, NF models have been applied to AD [164], [165].

However, a notable drawback is the absence of inherent dimensionality reduction. This limitation can be problematic in domains such as images, where the effective data dimensionality is much lower than the ambient dimensionality. Empirical studies have also shown that NF models can assign high likelihoods to anomalous instances [166]. Recent investigations suggest that this phenomenon may be related to the dominance of low-level features in likelihood computation and to architectural inductive biases [164], [167].

Their expressiveness and architectural flexibility can also be limited relative to other generative architectures. Despite

exact likelihood computation, generated image quality may remain inferior to that of GANs [168], and the high dimensionality of the latent space can complicate manipulation. Thus, exact likelihood is theoretically attractive, but it does not automatically translate into superior OOD or image-synthesis performance.

3) GRADIENT-BASED METHODS

The third notable branch of DL methods for OOD detection is gradient-based methods. Recent advancements have explored using gradient information from pre-trained models for OOD detection. These methods are promising because they offer strong performance and computational efficiency without requiring extensive fine-tuning. GradNorm [169] is based on the idea that ID data tends to have higher gradient magnitudes than OOD data, making gradient magnitude informative for OOD detection. ExGrad [170] further investigates this connection and relates the gradient norm to the energy score $E_\theta(x)$, showing that both share an "encoding-output" structure and that energy-like scores can be derived from gradients of the last and penultimate layers. More recently, GradOrth [171] uses an energy score derived from discriminative classifiers' gradients and applies orthogonal projection to improve OOD detection. Additionally, ViM [172] combines a class-agnostic score from the feature space with ID class-dependent logits and demonstrates the link between these components and the energy function. The recurring convergence of different methods toward energy-related scoring underscores the importance of energy functions in modern OOD detection.

Most gradient-based methods take the Kullback–Leibler divergence with respect to the uniform distribution $\mathbf{u}$ and measure a norm of the model gradients as the scoring function. For example, the GradNorm score is calculated as given in Equation (31).

$$S_{GN}(x) = \left\| \frac{\partial D_{\mathrm{KL}}\left(\mathbf{u} \parallel \operatorname{softmax}(f_\theta(x))\right)}{\partial W_L} \right\|_2^2 \tag{31}$$

where $f_\theta(x)$ denotes the logits and W_L denotes the weight matrix of layer L. Further reformulation in the original work yields the equivalent formula in Equation (32).

$$S_{GN}(x) = \frac{1}{T} \|\mathbf{h}\|_1 \sum_{k=1}^{C} \left| \frac{1}{C} - \mathbf{p}^{(k)} \right| \tag{32}$$

where $\mathbf{h}$ denotes the penultimate-layer encoding fed to the last layer of the network, $\mathbf{p}^{(k)}$ is the predicted probability for class k, and T is a temperature parameter. Following their notation, U denotes the part of the score characterized by $\mathbf{h}$, while V denotes the part characterized by the network output. In GradNorm, $U = \|\mathbf{h}\|_1$, $V = \sum_{k=1}^{C} \left| \frac{1}{C} - \mathbf{p}^{(k)} \right|$, and $S_{GN}(x) = \frac{1}{T} UV$. A related log-sum-exp energy score is given in Equation (33).

$$S_{\mathrm{Energy}}(x) = -T \log \sum_{k=1}^{C} e^{f^{(k)}(x)/T} \tag{33}$$

Under the same output-factor view, the pre-logit output term of the energy score is $V = \sum_{k=1}^{C} e^{f^{(k)}(x)/T}$. ExGrad proposed a similar composition where $U = 1$ and $V = \sum_{k=1}^{C} p^{(k)}(1 - p^{(k)})$. In addition, in ViM, the formula is a sum of two terms instead of a product, as shown in Equation (34).

$$S_{\mathrm{ViM}}(x) = \alpha \left\| x^{P^\perp} \right\| - \ln \sum_{i=1}^{C} e^{l_i} \tag{34}$$

Given their ability to provide a fine-grained view of model behavior, gradient-based methods remain an important component of modern OOD detection strategies, complementing generative models and reconstruction-based techniques.

4) GNNS

The final category of DL approaches for OOD detection consists of GNN methods. These methods leverage graph structure for AD when data are naturally represented as entities and relationships. Such settings arise in domains including social networks, biological networks, and fraud detection systems, where irregular behavior may appear through node attributes, edge patterns, or higher-order graph structure.

Recent graph OOD research has moved beyond using GNNs merely as generic encoders. Several works combine graph representation learning with predictive uncertainty, distribution-shift modeling, test-time adaptation, energy scoring, or robustness-oriented training [12], [13], [14], [173], [174], [175], [176]. In parallel, spectral and higher-order GNN studies highlight why graph representations must remain stable and discriminative under heterophily, deep propagation, and hypergraph structures [15], [16], [17], [177]. This subsection therefore introduces GNNs as graph-aware backbones, while Section IV provides the more detailed synthesis of graph OOD and graph AD methods.

a: CHEBNET

GNNs [178] provide a principled way to model relationships within graph-structured data. ChebNet, short for Chebyshev Network [53], is a representative spectral GNN architecture based on Chebyshev polynomials for graph signal processing [179]. ChebNet approximates graph convolutions using Chebyshev polynomials [180], defining graph convolutional filters in the spectral domain as shown in Equation (35).

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}) \tag{35}$$

Here, $g_\theta(\Lambda)$ is the filter, Λ is the diagonal matrix of eigenvalues of the graph Laplacian, $\tilde{\Lambda}$ denotes the rescaled eigenvalue matrix, T_k is the Chebyshev polynomial of order k, and $\theta \in \mathbb{R}^K$ is the vector of Chebyshev coefficients (filter parameters). This parametrization allows for localized filters.

The filters can be efficiently applied using a recursive formulation without explicitly computing eigenvectors,

as shown in Equation (36).

$$y = g_\theta(L)x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x \tag{36}$$

Here, $\tilde{L}$ denotes the rescaled graph Laplacian and $T_k(\tilde{L})$ applies the Chebyshev polynomial recursion on it. This formulation has linear complexity in the number of edges and efficiently captures localized graph patterns. Compared with earlier graph convolutional methods [181], [182], [183], ChebNet offers a practical balance between spectral filtering and local neighborhood computation.

In the broader context of GNNs, ChebNet shares commonalities with *Graph Convolutional Network* (GCN). While both architectures aim to extend convolutional operations to graph-structured data, ChebNet takes a distinctive spectral approach by leveraging Chebyshev polynomials. This approach enables ChebNet to extract information from a broader range of neighboring nodes, potentially enhancing its ability to model intricate graph structures. ChebNet laid the groundwork for subsequent developments in GCNs [42], streamlining the framework while capitalizing on the efficiency of Chebyshev filter localization. The spatial perspective of GCNs complements the spectral viewpoint of ChebNets.

b: GRAPH CONVOLUTIONAL NETWORKS

GCNs [42] are neural networks designed to work directly on graph-structured data. They were introduced to extend convolutional ideas to irregular domains such as social networks, knowledge graphs, and molecular graphs.

The key motivation behind GCNs is to develop neural network models that can efficiently leverage both graph structure and node feature information for tasks such as node classification and link prediction. Standard neural network models do not inherently preserve graph structure [178], [184], while traditional graph-based semi-supervised methods such as label propagation [185] often underutilize feature information. GCNs address this by using a localized first-order approximation of spectral graph convolutions as a propagation model, given in Equation (37).

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{37}$$

Here, $\tilde{A} = A + I$ is the adjacency matrix with added self-connections, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ is the corresponding degree matrix, $H^{(l)}$ is the activation matrix in the l^{th} layer, $W^{(l)}$ is a layer-specific trainable weight matrix, and $\sigma(\cdot)$ denotes an activation function. This propagation model allows aggregated neighborhood information to be combined with the current node features. By stacking multiple such localized convolutional layers, GCNs can learn hierarchical representations of graph-structured data for effective semi-supervised learning.

Zhong et al. [186] propose a new perspective on AD with only video-level labels, formulating it as a supervised learning problem with noise labels. They develop a framework that alternately trains an action classifier and a ''label noise cleaner'' based on a GCN. The GCN propagates information between high-confidence and low-confidence snippets to reduce label noise. After several iterations, the trained action classifier can directly predict anomalies in test videos without any extra post-processing.

c: GRAPH ATTENTION NETWORKS

The *Graph Attention Network* (GAT) [43] is a neural architecture for graph-structured data that adopts a self-attention strategy [187]. By using masked self-attention layers, GATs assign distinct weights to neighboring nodes without requiring computationally intensive matrix decompositions or full prior knowledge of the graph structure. This design improves the flexibility of neighborhood aggregation.

The core of the GAT framework lies in its graph attentional layer, which computes attention coefficients between a node and its neighbors using a shared attentional mechanism, specifically a single-layer feedforward neural network. These coefficients denote the importance of each neighbor's features to the central node. Subsequently, a nonlinearity is applied to a linear combination of the neighbors' features based on the attention coefficients, yielding the final output features for the node. The fundamental components of GATs include operations on node feature vectors, denoted as $\mathbf{h}_i \in \mathbb{R}^F$. The architecture learns a shared attentional mechanism, represented by the function $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \to \mathbb{R}$. Attention coefficients, indicating the importance of node j's features to node i, are computed in Equation (38).

$$e_{ij} = a(\mathbf{W}\mathbf{h}_i, \mathbf{W}\mathbf{h}_j) \tag{38}$$

The attention coefficients are normalized across the neighborhood $\mathcal{N}_i$ using Equation (39).

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e_{ik})} \tag{39}$$

The node features are then updated by calculating a linear combination of neighboring node features, as shown in Equation (40).

$$\mathbf{h}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\mathbf{h}_j\right) \tag{40}$$

To further stabilize learning, GATs incorporate multi-head attention with multiple independent attention mechanisms. The model can accommodate variable-sized neighborhoods, undirected and directed graphs, inductive learning, and unordered node sets. Its computational complexity is comparable to that of GCN, and its architecture can be parallelized across nodes and edges.

d: GRAPHSAGE

GraphSAGE [44] is a framework designed for inductive representation learning on large graphs. Traditional graph-based methods [188], [189], [190] often struggle with scalability and generalization to unseen nodes or graphs.

Algorithm 1 GraphSAGE Algorithm

Input:
Graph $G = (V, E)$
Feature vectors $x_v, \forall v \in V$
Number of iterations K
Aggregation functions $\text{AGG}_k, \forall k \in \{1, \ldots, K\}$
Weight matrices $W^{(k)}, \forall k \in \{1, \ldots, K\}$
Output:
Learnable parameters Θ
Final node embeddings $z_v, \forall v \in V$

Initialization:
$\Theta \leftarrow \{W^{(1)}, W^{(2)}, \ldots, W^{(K)}\}$
$h_v^{(0)} \leftarrow x_v, \quad \forall v \in V$
for $k = 1, \ldots, K$ **do**
  for $v \in V$ **do**
    $h_{\mathcal{N}(v)}^{(k)} \leftarrow \text{AGG}_k(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\})$
    $h_v^{(k)} \leftarrow \sigma(W^{(k)} \,\text{concat}(h_v^{(k-1)}, h_{\mathcal{N}(v)}^{(k)}))$
  end for
  $h_v^{(k)} \leftarrow h_v^{(k)} / \|h_v^{(k)}\|_2$ for all $v \in V$
end for
$z_v \leftarrow h_v^{(K)}, \quad \forall v \in V$
Compute Unsupervised Loss for each node $u \in V$:

$$J_{\mathbf{z}_u} = -\log(\sigma(\mathbf{z}_u^\top \mathbf{z}_v)) - Q\,\mathbb{E}_{v_n \sim p_n(v)}\left[\log(\sigma(-\mathbf{z}_u^\top \mathbf{z}_{v_n}))\right]$$

GraphSAGE addresses these challenges by employing a sampling and aggregation strategy.

The core idea of GraphSAGE is to learn a function that generates node embeddings by sampling and aggregating information from local neighborhoods. Unlike traditional methods that operate on the entire graph [184], [191], [192], GraphSAGE operates in a mini-batch fashion, allowing it to scale to large graphs while supporting inductive generalization to unseen nodes or evolving graph structures. By applying sampling and aggregation across multiple layers, GraphSAGE captures local community patterns and hierarchical node representations. The model can be trained using standard optimization techniques such as *Stochastic Gradient Descent* or Adam.

Let $G = (V, E)$ be a graph with nodes V and edges E. Let $h_v^{(l)}$ represent the embedding of node v at layer l. The algorithm concludes by calculating a loss function, $J_{\mathbf{z}_u}$, for each node u. This loss incorporates a *Noise-Contrastive Estimation* approach, where v denotes a positive context node sampled from the neighborhood of u, and v_n denotes a negative node sampled from a noise distribution $p_n(v)$. GraphSAGE's iterative aggregation process enables it to capture local structural information efficiently, facilitating scalability and inductive learning on large graphs. The GraphSAGE algorithm can be outlined as in Algorithm 1.

The preceding discussion positions GNNs as graph-aware representation backbones rather than complete OOD detectors by themselves. Their value for OOD detection comes from how message passing, neighborhood aggregation, attention, or spectral filtering can be paired with scoring rules such as uncertainty, reconstruction error, one-class distance, or energy. The next subsection summarizes the resulting methodological landscape and prepares the transition to the more detailed graph-aware synthesis in Section IV.

C. COMPREHENSIVE TAXONOMY

As discussed previously, OOD detection methods can be organized into two primary branches: traditional methods and DL methods. Figure 3 illustrates this organization, which has been widely recognized in several recent survey papers [4], [18], [72], [193].

The first branch of Figure 3 depicts traditional approaches to OOD detection, which rely on classical statistical techniques and distance-based metrics. Classical density estimation methods model the probability density of the data to identify anomalies, while distance-based models assess dissimilarities between data points to detect outliers [82], [90], [95], [100], [194]. One-class classification techniques learn a representation of the normal class and classify instances based on their adherence to this learned boundary [46], [118]. This taxonomy provides a structured view of the major classical techniques while also making their common limitations visible. In particular, many traditional methods struggle to capture complex interdependencies within data and become less scalable or less reliable on large-scale, high-dimensional datasets.

In contrast, the second branch of Figure 3 represents DL approaches that utilize neural architectures such as CNNs, recurrent neural networks, Transformers, and GNNs. Within this category, reconstruction models leverage reconstruction errors for OOD detection; representative methods include AEs and PCA [47], [195]. Generative models identify anomalies by measuring deviations from the learned distribution; notable examples include EBMs, VAEs, and GANs [48], [154]. The taxonomy separates graph backbones, such as ChebNet, GCN, GAT, and GraphSAGE [42], [43], [44], [53], from graph OOD and graph AD method families that build on these backbones. Recent graph OOD and graph AD studies extend basic architectures with uncertainty estimation, energy scoring, graph autoencoding, test-time adaptation, dynamic modeling, and robustness-oriented training [12], [13], [14], [33], [36], [176], [196], [197]. This placement makes the transition to Section IV explicit: GNNs are not treated only as another DL backbone, but as a setting in which classical OOD principles must be reformulated to account for relational structure.

Having established the landscape of both traditional and DL-based OOD detection methods, the next section shifts focus to GNN-based techniques specifically. While this section introduced both GNN backbones and recent graph OOD/AD method families, Section IV examines how these architectures are combined with classical OOD principles,

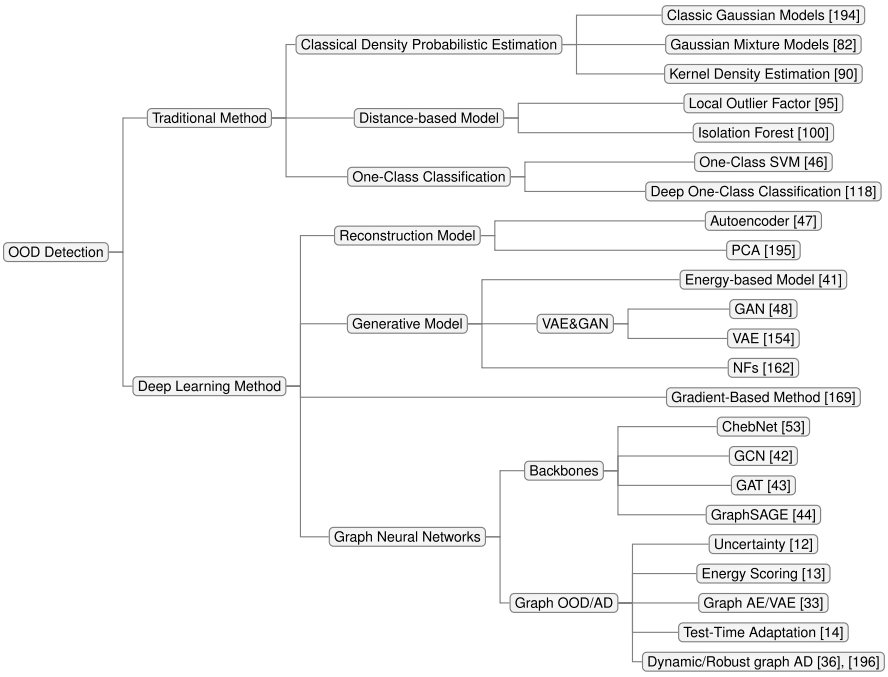

FIGURE 3. Taxonomy of representative OOD detection methods, grouped by traditional, DL-based, and graph-aware methodological families.

including uncertainty estimation, reconstruction, one-class classification, and energy-based scoring, to form specialized graph-aware OOD detection frameworks.

IV. GRAPH-AWARE OOD DETECTION WITH GNNS

Building on the taxonomy of traditional and DL methods presented in Section III, this section examines how classical OOD detection principles are reinterpreted within GNN frameworks. We first explain why relational structure changes the design of OOD scores, then broaden the coverage of recent graph OOD and graph AD methods before analyzing representative techniques, including DualGNN, OCGNN, and Energy-based GNN. These methods instantiate classical concepts such as representation refinement, one-class compactness, uncertainty scoring, reconstruction, and energy-based scoring on graph-structured data. We conclude with challenges and future directions.

A. FROM CLASSICAL PRINCIPLES TO GRAPH-AWARE OOD

Graph-based methods and GNNs have emerged as a relevant paradigm for AD and OOD detection [198], [199], [200]. Their role is not simply to provide stronger feature extractors, but to model relational dependencies that classical instance-wise detectors usually ignore. By aggregating information across nodes and neighborhoods, GNNs can encode local topology, broader structural context, and heterogeneous attributes, which is useful when anomaly or OOD evidence is distributed across multiple relational signals.

Among graph-based architectures, GNNs and GCNs have gained particular prominence. Well-known models such as GCN [42], GAT [43], and GraphSAGE [44] have been used to model complex relationships in domains such as transportation, healthcare, and social networks. DL techniques have also enabled objective functions for AD, including reconstruction losses, energy functions, and ELBO-style objectives, to be combined with GNN architectures [198], [199], [200], [201], [202]. At the same time, graph-based OOD detection is not solved simply by adding message passing. A common problem with plain GNNs for AD is that they may smooth anomalous nodes by incorporating information from their neighbors, making some outliers harder to identify. Tang et al. [177] investigate how anomalies affect the graph spectrum, showing that anomalies cause a "right-shift" effect where the spectral energy shifts from low to high frequencies.

Recent progress in graph representation learning further clarifies why graph OOD detection requires more than a direct transplant of classical OOD scores. Heterophilous graphs may violate the homophily assumptions implicitly used by many message-passing models; permutation-equivariant graph framelets provide one way to construct multiscale representations under such conditions [15]. Stability and generalization analyses of deep graph convolutional networks also show that depth, propagation strength, and graph spectrum affect whether learned representations preserve or erase discriminative signals [16]. In high-order relational settings, hypergraph and sheaf-based designs emphasize that high-pass components can be essential rather than merely noisy, which is especially relevant because anomalous or OOD nodes often appear as high-frequency deviations from their neighborhoods [17]. These studies are not OOD detectors by themselves, but they provide important architectural and theoretical context for designing graph

OOD methods that remain effective under heterophily, deep propagation, and high-order interactions.

1) RECENT GRAPH OOD AND GRAPH AD METHODS

Recent graph OOD research has expanded beyond generic GNN backbones toward detectors that explicitly model uncertainty, distribution shift, and graph-specific anomaly signals. Graph Posterior Networks use Bayesian predictive uncertainty for node classification, connecting graph OOD detection with probabilistic confidence estimation [12]. Learning on graphs with OOD nodes studies node-level OOD settings where abnormal nodes coexist with ID nodes in the same graph, highlighting the challenge of separating semantic novelty from neighborhood information [176]. GOOD-D investigates unsupervised graph OOD detection and emphasizes that graph OOD samples may differ in features, structure, or both [196]. GraphDE formulates graph debiased learning and graph OOD detection through a probabilistic generative framework with latent environments [203]. Energy-based GNN methods such as GNNSafe adapt the classical energy score to graph node classification and further use belief propagation to exploit relational dependencies [13], while NODESAFE addresses instability in node-level energy aggregation by bounding negative energy scores and mitigating logit shift [204]. GOODAT addresses test-time graph OOD detection, showing that adaptation strategies can be useful when the test distribution shifts after deployment [14]. Recent synthetic-exposure approaches such as GOLD further investigate how pseudo-OOD graph embeddings can be generated without auxiliary OOD data [205].

Graph AD studies provide complementary evidence for the same conceptual bridge. Deep graph autoencoders use reconstruction errors to identify abnormal attributed nodes [33], [206]; Graph VAE methods extend generative latent-variable reasoning to structured traces and attributed graphs [35]; and dual-discriminative GNNs address imbalanced graph-level anomaly detection by learning discriminative representations for rare abnormal graphs [207]. Other recent work explores few-shot transfer for graph anomaly detection [208], dynamic graph anomaly detection [197], and robustness of GNN-based anomaly detectors under adversarial graph perturbations [36]. Together, these studies show that recent graph OOD and graph AD methods are not limited to one-class or energy formulations; they also include uncertainty estimation, reconstruction, generative modeling, test-time adaptation, dynamic modeling, and adversarial robustness.

Viewed comparatively, these methods differ along four practical axes. First, detection granularity ranges from node-level uncertainty and OOD-node detection [12], [13], [176] to graph-level OOD and anomaly detection [196], [203], [207]. Second, the target shift may arise from node semantics, attributes, graph structure, or temporal evolution, so a method that is effective for feature novelty may not transfer directly to structural or temporal change. Third, the learning regime varies from unsupervised detection [33], [196] and synthetic OOD exposure [205] to adaptation at test time [14]. Finally, the resulting OOD evidence can be uncertainty-based [12], reconstruction-based [33], [206], latent-generative [35], [203], or energy-based [13], [204]. These distinctions clarify that graph OOD methods should be compared under matched detection units, shift mechanisms, supervision assumptions, and scoring objectives rather than treated as interchangeable GNN-based detectors.

2) HYPOTHESIS-GUIDED SYNTHESIS

The working hypotheses stated in the Introduction provide a compact way to organize the methods reviewed in this section. H1 is reflected by methods that first learn GNN representations and then apply a classical-style OOD score, such as uncertainty, reconstruction error, one-class distance, or energy. H2 is reflected by approaches that explicitly exploit graph topology, spectral behavior, temporal shifts, or neighborhood propagation, rather than treating graph nodes as independent samples. H3 motivates hybrid designs in which multiple classical principles are combined within the same graph learning framework. Under this view, DualGNN mainly illustrates representation refinement under graph structure, OCGNN illustrates one-class compactness in a learned graph embedding space, and energy-based GNN methods illustrate density-oriented scoring on graph-dependent logits. GEO, discussed in Section V, serves as an illustrative case study of H3.

Table 4 should therefore be read as a hypothesis-guided citation map rather than as an exhaustive benchmark comparison. With respect to H1, each left-to-right pairing indicates a classical OOD principle that has been transferred to, or reinterpreted within, a graph-aware setting. With respect to H2, the GNN-based column identifies adaptations that incorporate graph representation learning, message passing, topology-aware propagation, or graph-specific uncertainty and energy scoring. With respect to H3, rows that combine learned graph representations with one-class, reconstruction, generative, or energy-based objectives point to hybrid design patterns, while dashed entries mark families where direct GNN counterparts remain comparatively underdeveloped and may motivate future graph-aware formulations.

The following subsections then discuss representative GNN-based instantiations in more detail, focusing on how graph representation learning, one-class objectives, and energy-based scoring operationalize the H1–H3 links summarized in Table 4.

3) DUALGNN

DualGNN illustrates how graph representation refinement can support anomaly and OOD-related prediction tasks through complementary message-passing modules. Algorithm 2 summarizes this representative framework, which utilizes two GNN-based node prediction modules. Each module contains its own node embedding encoder (f) and node classifier (g).

TABLE 4. Conceptual mapping between non-GNN OOD method families and graph-aware adaptations.

Methodology (Non-GNN-based Group)			Adapted Methods Using GNNs (GNN-based Group)
Conventional Approaches	§ III-A1 Classic Density Estimation Probabilistic Models	Classic Gaussian Method [50], [51], [73]–[79]	[14], [173]
		Gaussian Mixture Model [75], [80]–[85]	[209]
		Kernel Density Estimation [86]–[94]	[173]
	§ III-A2 Distance-based Models	Local Outlier Factor [95]–[98]	—
		Isolation Forest [22], [100]–[109], [210]	—
	§ III-A3 One-Class Classification	One-class SVM [46], [70], [110]–[117]	[45]
		Deep One-Class Classification [70], [99], [118]	[211]
Deep Learning Approaches	§ III-B1 Reconstructive Models	Autoencoder [40], [86], [119]–[125], [197]	[206], [212]
		PCA [51], [126]–[138]	—
	§ III-B2 Generative Models	Energy-based [39], [41], [140]–[146]	[13], [211]
		VAE & GAN [34], [48], [149]–[156], [158], [196]	[33]–[35]
		NFs [160]–[168]	[213]
	§ III-B3 Gradient-based Methods	GradNorm [169], ExGrad [170], GradOrth [171], ViM [172]	—
	§ III-B4 Graph Neural Networks	—	ChebNet [42], [53], [178]–[181]
			GCN [42], [178], [184]–[186], [208], [12], [207], [214]
			GAT [43], [176], [187], [176]
			GraphSAGE [44], [184], [188]–[192], [214]

The first module acts as a primary GNN model that takes the initial node embedding matrix (X) and adjacency matrix (A) as inputs. It learns node embeddings by propagating messages from labeled nodes to the rest of the graph according to the input adjacency matrix. However, limited labeled nodes and noisy graph structure can restrict this module, since message propagation may remain concentrated around labeled neighborhoods and fail to reach a substantial portion of the graph.

To address this challenge, the second GNN module constructs a refined graph adjacency matrix using a fine-grained spectral clustering method based on the node embeddings produced by the first module. This refined adjacency matrix facilitates broader message propagation, enabling the induction of updated node embeddings and the construction of an auxiliary node classifier.

The two GNN modules are trained jointly, so the learned node embeddings are iteratively updated using both local information from the primary classifier and broader structural information from the auxiliary classifier.

Algorithm 2 Training Algorithm for the Dual GNN Learning Framework

Input: Given graph G with input feature matrix X
Adjacency matrix A
Labeled node set V_ℓ
Hyper-parameters α, K
Output: Learned model parameters Θ, Ω, $\widetilde{\Theta}$, $\widetilde{\Omega}$, Ψ

for iter = 1 **to** maxiters **do**
  $H = f_\Theta(X, A)$, compute node embeddings H
  $P = g_\Omega(H)$, compute nodes classification probabilities
  $S = h_\Psi(H)$, compute the fine-grained spectral clustering of nodes
  Construct adjacency matrix A_{sc} using S
  $\widetilde{H} = \widetilde{f}_{\widetilde{\Theta}}(H, A_{sc})$, compute node embeddings $\widetilde{H}$
  $\widetilde{P} = \widetilde{g}_{\widetilde{\Omega}}(\widetilde{H})$, compute nodes classification probabilities
  Calculate $\mathcal{L} = \mathcal{L}_{CE} + \widetilde{\mathcal{L}}_{CE} + \mathcal{L}_{sc}$
  Update model parameters Θ, Ω, $\widetilde{\Theta}$, $\widetilde{\Omega}$, Ψ to minimize $\mathcal{L}$ with gradient descent
end for

4) OCGNN

Wang et al. [45] introduced the *One-Class Graph Neural Network* (OCGNN) framework for AD on attributed networks. OCGNN combines GNN-based node embedding with a hypersphere learning objective that encloses normal nodes in the embedding space. Specifically, it jointly optimizes the GNN parameters and a hypersphere characterized by a center vector and radius, minimizing the volume of the sphere enclosing embeddings of normal training nodes. The anomaly score for a node is then determined by whether its embedding lies within this hypersphere. The objective function is based

on SVDD [215] and is given in Equation (41).

$$L_{\text{one-class}}(r, W) = \frac{1}{\beta K} \sum_{v_i \in V_{tr}} \left[\|g(X, A; W)_{v_i} - c\|_2^2 - r^2 \right]_+ + r^2 + \frac{\lambda}{2} \sum_{l=1}^{L} \|W^{(l)}\|_F^2 \quad (41)$$

where $g(X, A; W)$ is a GNN with parameters W, V_{tr} is the set of training nodes, $K = |V_{tr}|$, c is the hypersphere center vector, r is the hypersphere radius, $[z]_+ = \max(0, z)$, and β, λ are hyperparameters. The first term penalizes embeddings that fall outside the hypersphere, the second term minimizes the hypersphere volume, and the third term regularizes the network parameters. This loss function enables the simultaneous learning of useful graph embeddings and optimization of the one-class objective for AD, encapsulating the fundamental idea of extending hypersphere learning to graph data using GNNs.

The authors experiment with various GNN architectures, including GCN, GAT, and GraphSAGE, within the OCGNN framework. Their reported results on citation network benchmarks show improvements over several graph-embedding and AE-based baselines. The OCGNN framework is model-agnostic and extends classical one-class classifiers [216], including OCSVM, to non-Euclidean graph data. This reduces the need for hand-engineered graph features or statistics for AD.

5) ENERGY-BASED GNN

Many GNN models primarily focus on ID performance [173], [174], [175], often leaving test-time OOD risk insufficiently addressed. This motivates OOD methods that account for graph interdependence. The energy-based graph model in [13] adapts energy scoring to node-level graph OOD detection and introduces an energy-based belief propagation scheme to incorporate graph topology, reporting up to a 17.0% AUROC gain over compared methods. The method can operate without additional OOD data, while also allowing OOD exposure when such data are available. OOD data can be generated through multi-graph scenarios, temporal splits in single graphs, or synthetic graph structures, features, and labels.

The energy model itself is based on an energy function directly derived from the predicted logits of a GNN classifier trained with standard supervised classification loss. For an input x, the energy function is given in Equation (42). This definition follows the same score convention used earlier: ID samples are encouraged to have lower energy, while higher energy provides stronger OOD evidence.

$$E(x, G_x; h_\theta) = -\log \sum_{c=1}^{C} \exp(h_\theta(x, G_x)[c]) \quad (42)$$

Here, G_x represents the ego-graph around x, h_θ is the GNN classifier or logit function, and $[c]$ indexes the c-th class logit. This expression corresponds to the standard log-sum-exp energy score with temperature set to $T = 1$. Additionally, an energy regularization loss can be introduced when OOD training data is available, as shown in Equation (43).

$$L_{\text{energy}} = \frac{1}{|I_s|} \sum_{i \in I_s} \text{ReLU}(\tilde{E}(x_i, G_{x_i}; h_\theta) - t_{in})^2 + \frac{1}{|I_o|} \sum_{j \in I_o} \text{ReLU}(t_{out} - \tilde{E}(x_j, G_{x_j}; h_\theta))^2 \quad (43)$$

Here, I_s and I_o denote ID and OOD training indices, respectively; $\tilde{E}$ denotes the energy score used in the regularizer; and t_{in} and t_{out} are ID and OOD energy margins, typically chosen with $t_{in} < t_{out}$ so that ID samples are encouraged to have lower energy than OOD samples.

Beyond static message passing, graph Transformer encoders have also been explored for node-level prediction, as discussed in [13]. Extending energy-based graph OOD objectives to Transformer-style graph encoders is therefore a possible direction for future work.

B. CHALLENGES AND FUTURE DIRECTIONS

OOD detection remains challenging despite advances in DL and AD techniques. Existing methods still face limitations in robustness, generalization, and practical deployment. This subsection highlights four challenges that affect OOD detection methods and discusses future research directions motivated by them.

1) DETECTION RELIABILITY AND FALSE POSITIVES

Since anomalies are often rare and highly diverse, identifying them reliably remains difficult [9]. In practice, many normal cases are falsely reported as anomalies, while complex and genuinely novel anomalies may be overlooked. Although numerous AD methods have been introduced over the years, recent techniques, particularly unsupervised approaches, may still yield high false positive rates on real-world datasets. Reducing false positives while preserving strong recall therefore remains one of the most important challenges in OOD detection, especially in safety-critical applications where missed anomalies may have serious consequences. A related future direction is to investigate more reliable confidence estimation, calibration, and thresholding strategies for OOD detection systems.

2) HIGH-DIMENSIONAL AND INTERDEPENDENT DATA

Anomalies may exhibit clear aberrant characteristics in low-dimensional settings but become obscured and difficult to distinguish in high-dimensional spaces. Detecting anomalies in lower-dimensional subspaces, derived from a small set of original or engineered features, is one possible solution, as illustrated by subspace-based methods [217], [218], feature selection techniques [219], [220], and related approaches. However, capturing the higher-order, nonlinear, and heterogeneous interactions among features in high-dimensional data remains a major challenge. Moreover,

ensuring that a transformed feature space still preserves anomaly-relevant information is itself difficult, given the ambiguous and diverse nature of anomalies. In addition, many AD problems involve instances with interdependencies, such as temporal, spatial, or graph-based relationships, which further complicate detection. These difficulties suggest that future work may benefit from representation learning methods that preserve anomaly-relevant information while modeling dependencies among samples more explicitly.

3) LIMITED LABELS AND REALISTIC EVALUATION PROTOCOLS

Due to the difficulty of collecting large-scale labeled anomaly data, fully supervised AD is often impractical, since it assumes sufficient and accurate labels for both normal and anomalous classes. For this reason, much prior work has focused on unsupervised approaches, which do not require labeled anomaly data. However, unsupervised methods do not have direct access to true anomaly labels and therefore depend heavily on assumptions about anomaly distributions. As a result, effectively utilizing limited labeled data to learn normal and anomalous representations remains an important challenge [221]. Semi-supervised and weakly supervised AD have therefore emerged as promising directions, especially in settings where only normal data and/or a small subset of labeled anomalies are available, and where some anomaly labels may themselves be inaccurate. A closely related future direction is to develop more realistic benchmark datasets and more transparent evaluation protocols, particularly for settings involving limited, noisy, or incomplete anomaly supervision.

4) BEYOND NATIVE GRAPH DATA

One major challenge in experimenting with non-graph datasets is the absence of an explicit relational structure to exploit. Unlike graph data, where nodes and edges provide topological information, non-graph data such as images, text, and tabular records do not naturally encode such interconnectedness. This makes it more difficult to transfer graph-based methodologies directly. For example, common techniques in graph machine learning, such as message passing, node embedding, and link prediction, cannot be applied without additional modeling assumptions. Instead, researchers often rely on more conventional approaches such as convolutional architectures, feature extraction, or clustering-based methods. A potentially useful research direction is therefore to investigate how graph-like structure might be induced implicitly in non-graph domains, so that insights from graph representation learning can be adapted more systematically. At the same time, this direction raises additional questions related to computational cost, stability, and interpretability.

Taken together, these challenges also reinforce one of the central perspectives of this survey. As summarized by the taxonomy in Figure 3 and Table 4, many GNN-based OOD methods reuse or reinterpret ideas from classical methods while extending them through modern DL and graph representation learning. This observation suggests that the relationship between classical and graph-based methods may be useful not only for organizing prior work, but also for motivating future research. In particular, classical methods that do not yet have clear GNN-based counterparts may still inspire new graph-aware formulations for AD and OOD detection. For example, methods such as LOF, KDE, Isolation Forest, and energy-based scoring may be further investigated as regularizers, auxiliary objectives, or differentiable scoring components within graph-based models. More broadly, the conceptual bridge emphasized in this survey may serve as a useful source of hypotheses for future hybrid methods that combine statistical interpretability with the representational strengths of modern GNNs.

V. ILLUSTRATIVE CASE STUDY: GEO

A. MOTIVATION AND SCOPE

The conceptual bridges identified in Sections III and IV, particularly between energy-based scoring, one-class classification, and GNN-based learning, motivate a natural H3-oriented question:

> *How can complementary classical OOD principles be combined within a graph representation learning framework?*

GEO provides one concrete instantiation of this hybrid-objective view by integrating GNN representation learning with energy-based scoring and one-class regularization. As illustrated in Figure 4, the framework starts from an ID graph split into *IND-Tr*, *IND-Val*, and *IND-Te* regions, learns a *node embedding in structure space*, and then evaluates OOD evidence through two complementary branches. The upper branch applies an *energy function* $E(x; f)$ and plots score *frequency* over the *negative energy* axis, where a *threshold* τ separates *in-distribution* from *out-of-distribution* samples. The lower branch imposes a one-class *hypersphere* constraint on the learned embeddings. These signals are then combined through *fusion* to support final *out-of-distribution detection*. Within the survey, this section uses GEO to illustrate how the conceptual bridge can be implemented and evaluated on representative graph benchmarks.

In this reference study, GEO obtains competitive OOD detection results relative to confidence-based and graph-energy baselines in several settings [211]. The following subsections detail the framework, datasets, and representative experimental results.

B. GEO FRAMEWORK

This subsection summarizes GEO as a graph-aware hybrid objective for OOD detection. Following the visual flow in Figure 4, GEO first learns graph-aware node representations from the ID graph, then routes these representations into an energy-based branch and a one-class geometric branch. The energy branch models distributional evidence through the *energy function* and *threshold* mechanism, while the

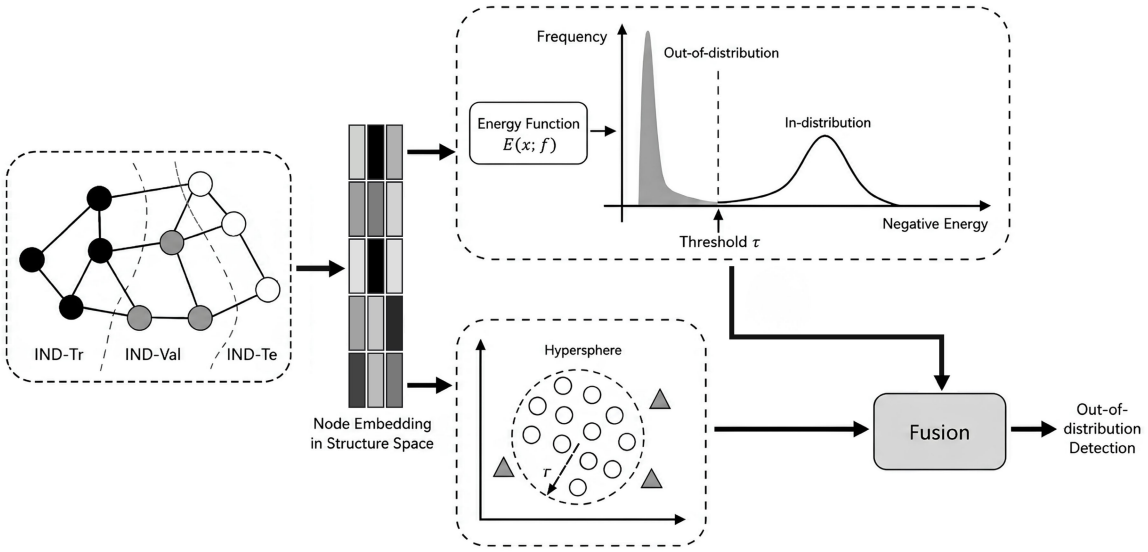

FIGURE 4. Overview of the GEO framework, which maps ID graph splits into *node embedding in structure space*, combines an *energy function* branch with a one-class *hypersphere* branch, and uses their *fusion* for OOD detection [211].

geometric branch constrains ID embeddings inside a compact *hypersphere*. This organization makes explicit how the H3 perspective is instantiated in practice by combining supervised graph representation learning, density-related energy scoring, and geometric compactness constraints.

1) GRAPH ENCODER AND STRUCTURE-SPACE EMBEDDING

The GEO framework begins with an input graph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of edges. The experimental split follows the intuition shown in Figure 4: *IND-Tr* nodes are used for training, *IND-Val* nodes support validation, and *IND-Te* nodes represent the ID portion of the test graph. Each node $v \in V$ is equipped with a feature vector $x_v \in \mathbb{R}^d$. The GNN encoder, denoted as f_θ with learnable parameters θ, takes the graph structure and the node features $\mathbf{X}$ as input to generate latent node embeddings $H = f_\theta(G, \mathbf{X})$, corresponding to the *node embedding in structure space* block in Figure 4. A classifier head h_θ then maps node embeddings to logits $Z = h_\theta(H)$. For notational compactness, $h_\theta(v, G_v)$ denotes the composed encoder–classifier logits for node v and its ego-graph G_v.

2) ENERGY AND ONE-CLASS OBJECTIVES

The upper branch in Figure 4 applies an *energy function* to the logits from the GNN classifier trained with supervised classification loss. The free energy function $E(v, G_v; h_\theta)$ is expressed as the negative log-sum-exp of the predicted logits over all classes, as previously formulated in Equation (42). Lower energy values correspond to ID samples, whereas higher values are used as stronger OOD evidence; equivalently, the figure plots score *frequency* over the *negative energy* axis, where a *threshold* τ separates *in-distribution* from *out-of-distribution* regions. When OOD data is available during training, an additional energy regularization loss L_{energy} (as formulated in Equation (43)) can be incorporated to push ID and OOD samples apart in the energy space. In parallel, the lower branch learns a tight *hypersphere* in the embedding space using the one-class loss $L_{\text{one-class}}$ (Equation (41)), enclosing the ID data and penalizing embeddings that fall outside the sphere.

3) TRAINING OBJECTIVE

The total loss combines the supervised negative log-likelihood (NLL), energy regularization, and one-class components as a weighted sum in Equation (44).

$$L_{\text{total}} = L_{\text{NLL}} + \alpha L_{\text{energy}} + \beta L_{\text{one-class}} \tag{44}$$

Here, L_{NLL} is the supervised node-classification loss computed from the GNN logits and node labels, L_{energy} enforces separation between ID and OOD samples in the energy space, and $L_{\text{one-class}}$ tightens the learned embedding space around the ID data. The hyperparameters α and β control the relative contributions of the energy and one-class terms.

4) MODEL-AGNOSTIC TRAINING LOOP

The framework is not tied to a specific GNN backbone and can use architectures such as GCN or GAT as the underlying encoder. During training, the model iteratively updates the GNN parameters, classifier head, and hypersphere parameters using the combined loss in Equation (44). OOD scores are then computed for test nodes or instances that were not used during training. Algorithm 3 summarizes this training and scoring procedure.

5) INFERENCE MECHANISM

During testing, the trained model produces OOD evidence from the energy-shaped logits and the compact embedding geometry. Conceptually, this corresponds to the *fusion* block

Algorithm 3 OOD Detection Framework

Input: Graph $G = (V, E)$ with node features $\mathbf{X}$ and partial labels Y
GNN encoder f_θ and classifier head h_θ
One-class hypersphere parameters: center c, radius r
Hyperparameters: α, β
Output: OOD score s for test nodes

Initialize f_θ, h_θ, c, r
for number of training iterations **do**
  Compute embeddings $H = f_\theta(G, \mathbf{X})$
  Compute logits $Z = h_\theta(H)$ and energy scores $s_E(v) = E(v, G_v; h_\theta)$
  Calculate $L_{\text{NLL}}(Y, Z)$
  Calculate L_{energy} using s_E
  Calculate $L_{\text{one-class}}(H, c, r)$
  $L_{\text{total}} \leftarrow L_{\text{NLL}} + \alpha L_{\text{energy}} + \beta L_{\text{one-class}}$
  Update f_θ, h_θ, c, r by descending ∇L_{total}
end for
Compute OOD score for each test node v:
$s(v) = E(v, G_v; h_\theta)$

in Figure 4, where the energy branch and the hypersphere-shaped representation jointly support the final *out-of-distribution detection* decision. In the reported evaluation, we use the standard energy score $E(v, G_v; h_\theta)$ as the primary OOD metric $s(v)$. The rationale for optimizing three distinct losses during training, while relying on a compact inference score, follows from the distinction between representation learning and anomaly scoring. During training, L_{NLL} encourages class separability, $L_{\text{one-class}}$ imposes a geometric boundary on ID representations, and L_{energy} encourages ID/OOD separation in the energy space. Once this regularized space is established, a new testing instance can be evaluated directly with the energy score, which provides a density-related signal shaped by the training losses. Higher energy scores indicate stronger OOD evidence under this scoring convention.

C. EXPERIMENTAL SETUP

This subsection describes the experimental setting used in the GEO case study. We first summarize the graph datasets, then define the OOD evaluation protocols and metric, and finally report the implementation details for the representative comparison.

1) DATASETS

We conduct experiments on four graph benchmarks: three medium-scale node-classification datasets with synthetic OOD protocols and one large-scale citation network with a realistic temporal shift. Table 5 reports the dataset statistics, keeping dataset scale separate from the OOD protocol design.

For `Cora`, `Amazon-Photo`, and `Coauthor-CS`, Table 5 reports the common undirected edge counts; implementations that explicitly store reverse edges may report approximately doubled edge counts. The `ogbn-arxiv` benchmark is reported as a directed citation graph following the Open Graph Benchmark statistics.

2) EVALUATION PROTOCOLS AND METRICS

To evaluate OOD detection under both synthetic and realistic settings, we use two protocols:

- **Synthetic Perturbations:** We apply this protocol to `Cora`, `Amazon-Photo`, and `Coauthor-CS`. OOD scenarios are synthesized via: (i) *structure manipulation* (generating OOD structures using the Stochastic Block Model), (ii) *feature interpolation* (random feature-space mixtures), and (iii) *label leave-out* (treating held-out classes as OOD at test time).
- **Real-world Temporal Shift:** For the `ogbn-arxiv` dataset, we follow the official OGB split protocol where papers published before 2018 serve as ID data, and those from 2019 onward are treated as OOD. This represents a realistic semantic shift in a dynamic citation network.

For each dataset, a limited fraction of node labels is used for supervised training, while another portion is reserved for validation. The remaining unperturbed/in-time nodes constitute the ID testing set. We use AUROC as the primary evaluation metric to measure the model's ability to distinguish between ID and OOD instances. Because this section is intended as an illustrative case study rather than a dedicated benchmark, complementary measures such as FPR@95TPR, AUPR, and AURC are not reproduced in this representative comparison.

3) IMPLEMENTATION DETAILS

All experiments were implemented using the PyTorch and PyTorch Geometric frameworks. The computational tasks were executed on a workstation equipped with an NVIDIA RTX 4060 GPU (8GB VRAM) and 16GB RAM, which was sufficient for the benchmark-scale graph experiments reported in this case study.

For model optimization, we employed the Adam optimizer with a learning rate of 0.01 and a weight decay of 0.01. Across all GNN backbones and scoring variants (GCN, GAT, and the softmax-based GEN variant [38]), the architecture was standardized to 2 message-passing layers with a hidden dimension size of 64. To mitigate overfitting, a dropout rate of 0.5 and batch normalization were applied. The models were trained for up to 200 epochs with an early stopping mechanism based on the validation loss to select the optimal weights.

D. REPRESENTATIVE RESULTS

As shown in Table 6, the representative evaluation covers ten OOD detection settings across four datasets: nine synthetic settings on `Cora`, `Amazon-Photo`, and `Coauthor-CS`, together with one realistic temporal-shift setting on `ogbn-arxiv`. Entries marked with "-" indicate configurations where training did not converge under the

TABLE 5. Dataset statistics used in the GEO case study.

Dataset	Graph type	Nodes	Edges	Features	Classes
Cora [65]	Citation	2,708	5,429	1,433	7
Amazon-Photo [66]	Co-purchase	7,650	119,081	745	8
Coauthor-CS [66]	Co-authorship	18,333	81,894	6,805	15
ogbn-arxiv [67]	Citation	169,343	1,166,243	128	40

TABLE 6. OOD detection performance measured by **AUROC** (↑) on Cora, Amazon-Photo, and Coauthor-CS under Structure manipulation (S), Feature interpolation (F), and Label leave-out (L), and on ogbn-arxiv under temporal shift. The strong graph energy baseline [13] is highlighted in purple; the best GEO variants are highlighted in teal; other black entries denote evaluated GEO configurations.

Backbone	Score Propagation	Energy Loss	One-Class Loss	Cora			Amazon-Photo			Coauthor-CS			ogbn-arxiv
				S	F	L	S	F	L	S	F	L	
MSP	No	No	No	71.54	85.20	91.15	78.67	97.45	42.38	94.63	96.59	93.96	63.91
GCN	No	No	Yes	70.98	86.26	91.35	98.30	97.97	93.65	96.19	97.87	95.57	66.41
GCN	No	Yes	Yes	46.68	49.93	67.62	71.69	76.50	73.25	80.46	93.23	85.36	53.64
GCN	Yes	No	Yes	77.62	93.40	-	98.66	98.53	**97.48**	99.61	99.66	97.01	71.06
GCN	Yes	Yes	No	**90.31**	**95.34**	**91.35**	**99.82**	**99.63**	93.65	**99.99**	**99.97**	**97.83**	**74.77**
GCN	Yes	Yes	Yes	90.24	**95.51**	-	98.89	**98.84**	96.94	**99.94**	99.92	97.10	**86.36**
GEN	No	No	Yes	70.99	86.08	91.35	99.20	97.32	93.75	96.13	97.84	95.50	66.37
GEN	No	Yes	Yes	75.38	88.06	71.01	99.84	98.45	90.57	98.25	99.04	91.23	77.71
GEN	Yes	No	Yes	87.12	93.23	92.69	98.65	98.52	97.47	99.62	99.64	97.04	65.08
GEN	Yes	Yes	No	90.25	95.43	91.50	98.86	98.68	-	**99.94**	**99.92**	**97.12**	77.62
GEN	Yes	Yes	Yes	90.18	95.49	91.48	98.91	98.84	97.04	**99.94**	**99.92**	97.08	78.35
GAT	No	No	Yes	75.64	89.43	92.74	99.98	98.42	**98.05**	97.08	98.65	94.97	55.06
GAT	No	Yes	Yes	74.42	89.58	59.63	**99.99**	98.52	91.56	98.28	99.21	74.83	75.78
GAT	Yes	No	Yes	86.19	94.26	**94.44**	98.69	98.41	97.61	99.51	99.71	96.75	54.07
GAT	Yes	Yes	No	87.15	94.44	92.68	98.74	98.50	96.88	99.85	99.88	96.17	69.39
GAT	Yes	Yes	Yes	87.22	94.82	92.75	98.74	98.49	96.96	99.86	99.88	96.10	76.30

given hyperparameters. The graph-based energy baseline achieves the best AUROC in 5 of the 10 settings, specifically on Cora (structure manipulation), Amazon-Photo (feature interpolation), and all three Coauthor-CS OOD protocols.

GEO configurations achieve the best AUROC in the remaining 5 settings, suggesting that the design space combining score propagation, energy regularization, and one-class compactness can be useful across several representative benchmarks. Notably, on the ogbn-arxiv dataset, which represents a challenging real-world temporal shift, the fully equipped variant (GCN with Score Propagation, Energy Loss, and One-Class Loss) attains an AUROC of **86.36%**, the best result among the compared methods. This exceeds the strongest baseline result of 74.77% and is consistent with the H3 view that energy-based and geometric constraints can provide complementary graph OOD signals.

Furthermore, adding the one-class loss to the GCN backbone with energy loss yields the top result on Cora (feature interpolation) with an AUROC of 95.51%. Additionally, three GAT-based variants achieve the best AUROC on Amazon-Photo (label leave-out), Amazon-Photo (structure manipulation), and Cora (label leave-out). In settings where GEO variants do not rank first, the closest gap between a GEO variant and the top score is 0.93% AUROC.

Overall, while no single architecture universally dominates all ten settings, the results indicate competitive OOD performance across datasets. For the purpose of this survey, the case study illustrates how classical scoring and geometric principles can be combined with graph representation learning, while also showing that the benefit is setting-dependent rather than universal.

VI. CONCLUSION

This survey examined OOD detection from the perspective of the conceptual relationship between classical machine learning methods and graph-based DL approaches. By organizing the literature into classical and DL-based branches, and by further analyzing how graph-based methods extend or reinterpret earlier statistical and geometric principles, the survey provides a unified view of the field. In particular, it highlights that OOD detection is not only a subproblem of AD, but also a distinct reliability problem that becomes especially important in complex, high-dimensional, and relational settings.

A central insight of this survey is that many modern GNN-based OOD detection methods can be understood not as isolated departures from classical approaches, but as graph-aware reformulations of long-standing ideas such as density estimation, distance-based scoring, one-class classification, and energy-based modeling. From this perspective, graph-based learning contributes primarily by providing expressive representations and relational inductive biases that make these principles applicable to structured and interdependent

data. The taxonomy and comparisons presented in this paper therefore suggest that the connection between classical and graph-based methods is not merely historical, but conceptually informative for understanding the current landscape of OOD detection. This analysis is consistent with the three working hypotheses introduced in the paper: classical OOD scores can be transferred into graph-aware representation spaces (H1), graph-specific structure should shape OOD scoring (H2), and hybrid objectives can combine complementary statistical, geometric, and relational signals (H3).

The GEO case study was included in this spirit as an illustrative example showing that the proposed conceptual bridge can also be instantiated in practice. Taken together, the analysis in this survey suggests that future progress in OOD detection may benefit from more principled integration of classical statistical reasoning, modern deep representation learning, and graph-based modeling. In this sense, the survey contributes a synthesized perspective on how these research directions relate to one another and why their interaction matters for reliable OOD detection in real-world applications.

ACKNOWLEDGMENT

This research is funded by the Murata Science Foundation under Grant No. 24VH09. The authors acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, for its support of this study. They also extend their sincere thanks to Tai T. Phan, Viet Q. Nguyen, Thai M. Nguyen, Khanh T. H. Nguyen, and Vinh Q. Vu from URA Research Group, HCMUT, for carefully verifying the references against authoritative official sources. Their assistance helped ensure the accuracy, reliability, and scholarly rigor of this survey.

REFERENCES

[1] M. Kukar, "Transductive reliability estimation for medical diagnosis," *Artif. Intell. Med.*, vol. 29, nos. 1–2, pp. 81–106, Sep. 2003.

[2] D. Dai and L. V. Gool, "Dark model adaptation: Semantic image segmentation from daytime to nighttime," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3819–3824.

[3] A. Elliott, M. Cucuringu, M. M. Luaces, P. Reidy, and G. Reinert, "Anomaly detection in networks with application to financial transaction networks," 2019, *arXiv:1901.00402,*.

[4] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *Int. J. Comput. Vis.*, vol. 132, no. 12, pp. 5635–5662, Jun. 2024.

[5] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," 2016, *arXiv:1606.06565*.

[6] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[7] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Sep. 2020.

[8] Q. Gao and P. Ma, "Graph neural network and context-aware based user behavior prediction and recommendation system research," *Comput. Intell. Neurosci.*, vol. 2020, Nov. 2020, Art. no. 8812370, doi: 10.1155/2020/8812370.

[9] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, Mar. 2021.

[10] S. Lu, Y. Wang, L. Sheng, L. He, A. Zheng, and J. Liang, "Out-of-distribution detection: A task-oriented survey of recent advances," *ACM Comput. Surv.*, vol. 58, no. 2, pp. 1–39, Sep. 2025.

[11] H. Qiao, H. Tong, B. An, I. King, C. Aggarwal, and G. Pang, "Deep graph anomaly detection: A survey and new perspectives," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 9, pp. 5106–5126, Sep. 2025.

[12] M. Stadler, B. Charpentier, S. Geisler, D. Zügner, and S. Günnemann, "Graph posterior network: Bayesian predictive uncertainty for node classification," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst.*, 2021.

[13] Q. Wu, Y. Chen, C. Yang, and J. Yan, "Energy-based out-of-distribution detection for graph neural networks," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023.

[14] L. Wang, D. He, H. Zhang, Y. Liu, W. Wang, S. Pan, D. Jin, and T.-S. Chua, "GOODAT: Towards test-time graph out-of-distribution detection," in *Proc. 38th AAAI Conf. Artif. Intell. 36th Conf. Innov. Appl. Artif. Intell. 14th Symp. Educ. Adv. Artif. Intell.*, 2024.

[15] J. Li, R. Zheng, H. Feng, M. Li, and X. Zhuang, "Permutation equivariant graph framelets for heterophilous graph learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 11634–11648, Sep. 2024.

[16] G. Yang, M. Li, H. Feng, and X. Zhuang, "Deeper insights into deep graph convolutional networks: Stability and generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 48, no. 2, pp. 1707–1719, Feb. 2026.

[17] M. Li, Y. Fang, D. Shen, H. Feng, X. Zhuang, K. Xia, and P. Lio, "High-pass matters: Theoretical insights and sheaflet-based design for hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 40, pp. 23039–23046.

[18] R. Xu and K. Ding, "Large language models for anomaly and out-of-distribution detection: A survey," in *Proc. Findings Assoc. Comput. Linguistics, NAACL*, Apr. 2025, pp. 6007–6027.

[19] F. Che, Q. Z. Ahmed, F. A. Khan, and P. I. Lazaridis, "Anomaly detection based on generalized Gaussian distribution approach for ultra-wideband (UWB) indoor positioning system," in *Proc. 26th Int. Conf. Autom. Comput. (ICAC)*, Sep. 2021, pp. 1–5.

[20] O. Rippel, P. Mertens, E. König, and D. Merhof, "Gaussian anomaly detection by modeling the distribution of normal data in pretrained deep features," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[21] M. Jiang, S. Han, and H. Huang, "Anomaly detection with score distribution discrimination," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2023, pp. 984–996.

[22] H. Xu, G. Pang, Y. Wang, and Y. Wang, "Deep isolation forest for anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12591–12604, Dec. 2023.

[23] M. Hejazi and Y. P. Singh, "One-class support vector machines approach to anomaly detection," *Appl. Artif. Intell.*, vol. 27, no. 5, pp. 351–366, May 2013.

[24] E. H. M. Pena, M. V. O. de Assis, and M. L. Proença, "Anomaly detection using forecasting methods ARIMA and HWDS," in *Proc. 32nd Int. Conf. Chilean Comput. Sci. Soc. (SCCC)*, Nov. 2013, pp. 63–66.

[25] J. Souza, E. Paixão, F. Fraga, L. Baroni, R. F. S. Alves, K. Belloze, J. dos Santos, E. Bezerra, F. Porto, and E. Ogasawara, "REMD: A novel hybrid anomaly detection method based on EMD and ARIMA," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2024, pp. 1–8.

[26] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi, "Deep learning for time series anomaly detection: A survey," *ACM Comput. Surveys*, vol. 57, no. 1, pp. 1–42, Oct. 2024.

[27] C. Tang, L. Xu, B. Yang, Y. Tang, and D. Zhao, "GRU-based interpretable multivariate time series anomaly detection in industrial control system," *Comput. Secur.*, vol. 127, Apr. 2023, Art. no. 103094.

[28] K. Zhan, C. Wang, X. Zheng, C. Kong, G. Li, W. Xin, and L. Liu, "Seq2Seq-based GRU autoencoder for anomaly detection and failure identification in coal mining hydraulic support systems," *Sci. Rep.*, vol. 15, no. 1, p. 542, Jan. 2025.

[29] M. Rieza Fachrezi, A. Firman Ihsan, and W. Astuti, "Anomaly detection using LSTM-based deep learning on natural gas pipeline operational data," in *Proc. 12th Int. Conf. Inf. Commun. Technol. (ICoICT)*, Aug. 2024, pp. 500–506.

[30] M. N. Amin, L. Hubner, F. S. Bayram, and A. Jesser, "Real-time anomaly detection with LSTM-autoencoder network on microcontrollers for industrial applications," in *Proc. 8th Int. Conf. Graph. Signal Process.*, Jun. 2024, pp. 42–46.

[31] M. Abdallah, N. An Le Khac, H. Jahromi, and A. Delia Jurcut, "A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs," in *Proc. 16th Int. Conf. Availability, Rel. Secur.*, Aug. 2021, pp. 1–7.

[32] F. K. Stehle, W. Vandelli, F. Zahn, G. Avolio, and H. Fröning, "DeepHYDRA: A hybrid deep learning and DBSCAN-based approach to time-series anomaly detection in dynamically-configured systems," in *Proc. 38th ACM Int. Conf. Supercomputing*, New York, NY, USA, May 2024, pp. 272–285.
[33] K. Ding, J. Li, R. Bhanushali, and H. Liu, *Deep Anomaly Detection on Attributed Networks*, 2019, pp. 594–602.
[34] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Information Processing in Medical Imaging*. Cham, Switzerland: Springer, 2017, pp. 146–157.
[35] Z. Xie, H. Xu, W. Chen, W. Li, H. Jiang, L. Su, H. Wang, and D. Pei, "Unsupervised anomaly detection on microservice traces through graph VAE," in *Proc. ACM Web Conf.*, New York, NY, USA, Apr. 2023, pp. 2874–2884.
[36] X. Zheng, B. Wu, A. X. Zhang, and W. Li, "Improving robustness of GNN-based anomaly detection by graph adversarial training," in *Proc. Lang. Resour. Eval. Conf.*, May 2024, pp. 8902–8912.
[37] X. Liang, X. Wang, Z. Lei, S. Liao, and S. Z. Li, "Soft-margin softmax for deep classification," in *Proc. Neural Inf. Process.*, 2017, pp. 413–421.
[38] X. Liu, Y. Lochman, and C. Zach, "GEN: Pushing the limits of softmax-based out-of-distribution detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 23946–23955.
[39] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2017.
[40] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, and S. Vernekar, "Improving reconstruction autoencoder out-of-distribution detection with Mahalanobis distance," 2018, *arXiv:1812.02765*.
[41] W. Liu, X. Wang, J. D. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020.
[42] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017.
[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018.
[44] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, New York, NY, USA, 2017, pp. 1025–1035.
[45] X. Wang, B. Jin, Y. Du, P. Cui, Y. Tan, and Y. Yang, "One-class graph neural networks for anomaly detection in attributed networks," *Neural Comput. Appl.*, vol. 33, no. 18, pp. 12073–12085, Sep. 2021.
[46] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Proc. 13th Int. Conf. Neural Inf. Process. Syst.*, Cambridge, MA, USA, 1999, pp. 582–588.
[47] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," 2015, *arXiv:2408.13561*.
[48] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-based anomaly detection," 2018, *arXiv:1802.06222*.
[49] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA, 2017, pp. 6405–6416.
[50] N. A. Ahuja, I. J. Ndiour, T. Kalyanpur, and O. Tickoo, "Probabilistic modeling of deep features for out-of-distribution and adversarial detection," 2019, *arXiv:1909.11786*.
[51] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2016.
[52] H. Choi and E. Jang, "Generative ensembles for robust anomaly detection," 2019, *arXiv:1810.01392*.
[53] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA, 2016, pp. 3844–3852.
[54] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
[55] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[56] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[58] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proc. EMNLP Workshop BlackboxNLP, Analyzing Interpreting Neural Netw. NLP*, Nov. 2018, pp. 353–355.
[59] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Portugal, Sep. 2015, pp. 632–642.
[60] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, "CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proc. 33rd AAAI Conf. Artif. Intell. 31st Innov. Appl. Artif. Intell. Conf. 9th AAAI Symp. Educ. Adv. Artif. Intell.*, 2019.
[61] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3462–3471.
[62] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "MedMNIST v2—A large-scale lightweight benchmark for 2D and 3D biomedical image classification," *Scientific Data*, vol. 10, no. 1, p. 41, Jan. 2023.
[63] S. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. Chan, "KDD cup 1999 data," UCI Mach. Learn. Repository, Tech. Rep., 1999. [Online]. Available: https://doi.org/10.24432/C51C7N
[64] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.
[65] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, pp. 93–106, Sep. 2008.
[66] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," 2018, *arXiv:1811.05868*.
[67] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA, 2020.
[68] S. Gui, X. Li, L. Wang, and S. Ji, "GOOD: A graph out-of-distribution benchmark," in *Proc. Adv. Neural Inf. Process. Syst. 35*, U.S., 2022, pp. 2059–2073.
[69] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Process.*, vol. 99, pp. 215–249, Jul. 2014.
[70] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.
[71] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Stat. Anal. Data Mining, ASA Data Sci. J.*, vol. 5, no. 5, pp. 363–387, Oct. 2012.
[72] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, Jul. 2009.
[73] R. De Maesschalck, D. Jouan-Rimbaud, and D. Massart, "The Mahalanobis distance," *Chemometrics Intell. Lab. Syst.*, vol. 50, no. 1, pp. 1–18, 2000.
[74] J. Laurikkala, M. Juhola, and E. Kentala, "Informal identification of outliers in medical data," in *Proc. 5th Int. Workshop Intell. Data Anal. Med. Pharmacol.*, 2000, pp. 20–24.
[75] S. Bickel and T. Scheffer, "Estimation of mixture models using co-EM," in *Machine Learning: ECML 2005*. Berlin, Germany: Springer, 2005, pp. 35–46.
[76] M. Rosenblatt, "A central limit theorem and a strong mixing condition," *Proc. Nat. Acad. Sci. USA*, vol. 42, no. 1, pp. 43–47, Jan. 1956.
[77] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2022.
[78] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 7167–7177.
[79] G. Amit, M. Levy, I. Rosenberg, A. Shabtai, and Y. Elovici, "GLOD: Gaussian likelihood out of distribution detector," Tech. Rep., 2020.

[80] S. Roberts and L. Tarassenko, "A probabilistic resource allocating network for novelty detection," *Neural Comput.*, vol. 6, no. 2, pp. 270–284, Mar. 1994.

[81] C. M. Bishop, "Novelty detection and neural network validation," in *ICANN '93*, 1993, pp. 789–794.

[82] D. Reynolds, *Gaussian Mixture Models*. Boston, MA, USA: Springer, 2009, pp. 659–663.

[83] W. Jannah and D. R. Saputro, "Parameter estimation of Gaussian mixture models (GMM) with expectation maximization (EM) algorithm," *AIP Conf. Proc.*, vol. 2566, Sep. 2022, Art. no. 040002.

[84] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for Gaussian mixtures," *Neural Comput.*, vol. 8, no. 1, pp. 129–151, Jan. 1996.

[85] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[86] R. A. Davis, K.-S. Lii, and D. N. Politis, *Remarks on Some Nonparametric Estimates of a Density Function*. New York, NY, USA: Springer, 2011, pp. 95–100.

[87] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Cham, Switzerland: Springer, 2009.

[88] S. M. Piryonesi and T. E. El-Diraby, "Role of data analytics in infrastructure asset management: Overcoming data size and quality problems," *J. Transp. Eng., B, Pavements*, vol. 146, no. 2, Jun. 2020, Art. no. 04020022.

[89] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," *Proc. Roy. Soc. London. Ser. A, Containing Papers Math. Phys. Character*, vol. 83, no. 559, pp. 69–70, Nov. 1909.

[90] X. Qin, L. Cao, E. Rundensteiner, and S. Madden, "Scalable kernel density estimation-based local outlier detection over large data streams," Tech. Rep., 2019.

[91] E. Erdil, K. Chaitanya, and E. Konukoglu, "Unsupervised out-of-distribution detection using kernel density estimation," Tech. Rep., 2020.

[92] P. Lv, Y. Yu, Y. Fan, X. Tang, and X. Tong, "Layer-constrained variational autoencoding kernel density estimation model for anomaly detection," *Knowl.-Based Syst.*, vol. 196, May 2020, Art. no. 105753.

[93] S. Vempati, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Generalized RBF feature maps for efficient detection," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 2.1–2.11, doi: 10.5244/c.24.2.

[94] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. 21st Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1177–1184.

[95] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. 2000 ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.

[96] M. Sugiyama, "Chapter 38—Outlier detection," in *Introduction to Statistical Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 2016, ch. 3, pp. 457–468.

[97] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, *A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection*, pp. 25–36.

[98] E. Schubert, A. Zimek, and H.-P. Kriegel, "Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 190–237, Jan. 2014.

[99] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, Sep. 1962.

[100] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, May 2008, pp. 413–422.

[101] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.

[102] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *J. Comput. Graph. Statist.*, vol. 15, no. 1, pp. 118–138, Mar. 2006.

[103] P. Gopalan, V. Sharan, and U. Wieder, "PIDForest: Anomaly detection via partial identification," in *Proc. Adv. Neural Inf. Process. Syst.*, Red Hook, NY, USA: Curran Associates, 2019.

[104] D. Cortes, "Revisiting randomized choices in isolation forests," 2021, *arXiv:2110.13402*.

[105] M. Tokovarov and P. Karczmarek, "A probabilistic generalization of isolation forest," *Inf. Sci.*, vol. 584, pp. 433–449, Jan. 2022.

[106] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1479–1489, Apr. 2021.

[107] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "On detecting clustered anomalies using SCiForest," in *Machine Learning and Knowledge Discovery in Databases*. Berlin, Germany: Springer, 2010, pp. 274–290.

[108] J. Lesouple, C. Baudoin, M. Spigai, and J.-Y. Tourneret, "Generalized isolation forest for anomaly detection," *Pattern Recognit. Lett.*, vol. 149, pp. 109–119, Sep. 2021.

[109] M. Chater, A. Borgi, M. T. Slama, K. Sfar-Gandoura, and M. I. Landoulsi, "Fuzzy isolation forest for anomaly detection," *Proc. Comput. Sci.*, vol. 207, pp. 916–925, Sep. 2022.

[110] D. M. J. Tax and R. P. W. Duin, "Combining one-class classifiers," in *Multiple Classifier Systems*. Berlin, Germany: Springer, 2001, pp. 299–308.

[111] D. M. J. Tax and R. P. W. Duin, "Data domain description using support vectors," in *Proc. Eur. Symp. Artif. Neural Netw.*, 1999.

[112] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognit. Lett.*, vol. 20, nos. 11–13, pp. 1191–1199, Nov. 1999.

[113] D. M. J. Tax and R. P. W. Duin, "Uniform object generation for optimizing one-class classifiers," *J. Mach. Learn. Res.*, vol. 2, pp. 155–173, Mar. 2002.

[114] H. Yu, "Single-class classification with mapping convergence," *Mach. Learn.*, vol. 61, nos. 1–3, pp. 49–69, Nov. 2005.

[115] L. M. Manevitz and M. Yousef, "One-class svms for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, Mar. 2002.

[116] K.-L. Li, H.-K. Huang, S.-F. Tian, and W. Xu, "Improving one-class SVM for anomaly detection," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Mar. 2003, pp. 3077–3081.

[117] P.-Y. Hao, "Fuzzy one-class support vector machines," *Fuzzy Sets Syst.*, vol. 159, no. 18, pp. 2317–2336, Sep. 2008.

[118] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 4393–4402.

[119] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991.

[120] Y. Zhou, "Rethinking reconstruction autoencoder-based out-of-distribution detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7369–7377.

[121] Y. Yang, R. Gao, and Q. Xu, "Out-of-distribution detection with semantic mismatch under masking," in *Computer Vision—ECCV 2022*. Cham, Switzerland: Springer, 2022, pp. 373–390.

[122] R. Hinami, T. Mei, and S. Satoh, "Joint detection and recounting of abnormal events by learning deep generic knowledge," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Los Alamitos, CA, USA, Oct. 2017, pp. 3639–3647.

[123] Q. Liu, J. Xu, R. Jiang, and W. H. Wong, "Roundtrip: A deep generative neural density estimator," 2020, *arXiv:2004.09017*.

[124] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 481–490.

[125] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," 2016, *arXiv:1611.01704*.

[126] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, Jun. 2011.

[127] A. Mackiewicz and W. Ratajczak, "Principal components analysis (PCA)," *Comput. Geosci.*, vol. 19, no. 3, pp. 303–342, 1993.

[128] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *Proc. IEEE*, vol. 109, no. 5, pp. 756–795, May 2021.

[129] D. P. Bertsekas, "Chapter 1—Introduction," in *Constrained Optimization and Lagrange Multiplier Methods*. New York, NY, USA: Academic, 1982, ch. 1, pp. 1–94.

[130] S. Marukatat, "Tutorial on PCA and approximate PCA d approximate kernel PCA," *Artif. Intell. Rev.*, vol. 56, no. 6, pp. 5445–5477, Oct. 2022.

[131] B. Mnassri, E. M. El Adel, and M. Ouladsine, "Fault localization using principal component analysis based on a new contribution to the squared prediction error," in *Proc. 16th Medit. Conf. Control Autom.*, Jun. 2008, pp. 65–70.

[132] B. Mnassri, E. M. El Adel, and M. Ouladsine, "Reconstruction-based contribution approaches for improved fault diagnosis using principal component analysis," *J. Process Control*, vol. 33, pp. 60–76, Sep. 2015.

[133] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
[134] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern Recognit.*, vol. 40, no. 3, pp. 863–874, Mar. 2007.
[135] N. Kwak, "Principal component analysis based on L_1-norm maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1672–1680, Sep. 2008.
[136] M. H. Nguyen and F. D. L. Torre, "Robust kernel principal component analysis," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.*, 2008, pp. 1185–1192.
[137] D. Brauckhoff, K. Salamatian, and M. May, "Applying PCA for traffic anomaly detection: Problems and solutions," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2866–2870.
[138] K. Chowdhary and H. N. Najm, "Bayesian estimation of Karhunen–Loève expansions; A random subspace approach," *J. Comput. Phys.*, vol. 319, pp. 280–293, Aug. 2016.
[139] D. Wang, L. Ren, X. Sun, L. Gao, and J. Chanussot, "Nonlocal and local feature-coupled self-supervised network for hyperspectral anomaly detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 18, pp. 6981–6993, 2025.
[140] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
[141] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, vol. 5, 2009, pp. 448–455.
[142] Y. LeCun, S. Chopra, R. Hadsell, A. Ranzato, and F. J. Huang, "A tutorial on energy-based learning," Tech. Rep., 2006.
[143] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.
[144] A. Hyvärinen, "Estimation of non-normalized statistical models by score matching," *J. Mach. Learn. Res.*, vol. 6, pp. 695–709, Dec. 2005.
[145] J. Ngiam, Z. Chen, P. W. Koh, and A. Y. Ng, "Learning deep energy models," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, 2011, pp. 1105–1112.
[146] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, "Deep structured energy based models for anomaly detection," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 1100–1109.
[147] Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
[148] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *Proc. Int. Conf. Learn. Represent.*, 2020.
[149] M. Lucic, K. Kurach, M. Michalski, O. Bousquet, and S. Gelly, "Are GANs created equal? A large-scale study," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 698–707.
[150] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *Proc. Int. Conf. Learn. Represent.*, 2017.
[151] B. Dai and D. Wipf, "Diagnosing and enhancing VAE models," in *Proc. Int. Conf. Learn. Represent.*, 2019.
[152] R. Child, "Very deep VAEs generalize autoregressive models and can outperform them on images," in *Proc. Int. Conf. Learn. Represent.*, 2021.
[153] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, *An Introduction to Variational Methods for Graphical Models*. Cambridge, MA, USA: MIT Press, 1999, pp. 105–161.
[154] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
[155] X. Ran, M. Xu, L. Mei, Q. Xu, and Q. Liu, "Detecting out-of-distribution samples via variational auto-encoder with reliable uncertainty estimation," *Neural Netw.*, vol. 145, pp. 199–208, Jan. 2022.
[156] J. Schmidhuber, "Making the world differentiable: On using self supervised fully recurrent neural networks for dynamic reinforcement learning and planning in non-stationary environments," Forschungsberichte, Munich, Germany, Tech. Rep., 1990, pp. 1–26.
[157] J. Schmidhuber, "Generative adversarial networks are special cases of artificial curiosity (1990) and also closely related to predictability minimization (1991)," *Neural Netw.*, vol. 127, pp. 58–66, Jul. 2020.
[158] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow, "Adversarial autoencoders," 2015, *arXiv:1511.05644*.
[159] D. Wang, L. Gao, Y. Qu, X. Sun, and W. Liao, "Frequency-to-spectrum mapping GAN for semisupervised hyperspectral anomaly detection," *CAAI Trans. Intell. Technol.*, vol. 8, no. 4, pp. 1258–1273, Dec. 2023.
[160] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," 2014, *arXiv:1410.8516*.
[161] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *J. Mach. Learn. Res.*, vol. 22, Jan. 2021.
[162] I. Kobyzev, S. J. D. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3964–3979, Nov. 2021.
[163] E. G. Tabak and C. V. Turner, "A family of nonparametric density estimation algorithms," *Commun. Pure Appl. Math.*, vol. 66, no. 2, pp. 145–164, Feb. 2013.
[164] B. Nachman and D. Shih, "Anomaly detection with density estimation," *Phys. Rev. D*, vol. 101, no. 7, Apr. 2020, Art. no. 075042.
[165] L. Wellhausen, R. Ranftl, and M. Hutter, "Safe robot navigation via multi-modal anomaly detection," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1326–1333, Apr. 2020.
[166] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they Don't know?" in *Proc. Int. Conf. Learn. Represent.*, 2019.
[167] R. Schirrmeister, Y. Zhou, T. Ball, and D. Zhang, "Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21038–21049.
[168] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7327–7347, Nov. 2022.
[169] R. Huang, A. Geng, and Y. Li, "On the importance of gradients for detecting distributional shifts in the wild," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst.*, 2021.
[170] C. Igoe, Y. Chung, I. Char, and J. Schneider, "How useful are gradients for OOD detection really?" 2023, *arXiv:2205.10439*.
[171] S. Behpour, T. L. Doan, X. Li, W. He, L. Gou, and L. Ren, "GradOrth: A simple yet efficient out-of-distribution detection with orthogonal projection of gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 38206–38230.
[172] H. Wang, Z. Li, L. Feng, and W. Zhang, "ViM: Out-of-distribution with virtual-logit matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4911–4920.
[173] X. Zhao, F. Chen, S. Hu, and J.-H. Cho, "Uncertainty aware semi-supervised learning on graph data," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020.
[174] J. Ma, J. Deng, and Q. Mei, "Subgroup generalization and fairness of graph neural networks," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst.*, 2021.
[175] Q. Wu, H. Zhang, J. Yan, and D. Wipf, "Handling distribution shifts on graphs: An invariance perspective," in *Proc. Int. Conf. Learn. Represent.*, 2022.
[176] Y. Song and D. Wang, "Learning on graphs with out-of-distribution nodes," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2022, pp. 1635–1645.
[177] J. Tang, J. Li, Z. Gao, and J. Li, "Rethinking graph neural networks for anomaly detection," in *Proc. 39th Int. Conf. Mach. Learn.*, vol. 162, 2022, pp. 21076–21089.
[178] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, pp. 61–80, 2009.
[179] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
[180] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.
[181] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*.
[182] A. Coates and A. Y. Ng, "Selecting receptive fields in deep networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 2528–2536.
[183] K. Gregor and Y. LeCun, "Emergence of complex-like cells in a temporal product network with local receptive fields," 2010, *arXiv:1006.0448*.
[184] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, Aug. 2005, pp. 729–734.

[185] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Int. Conf. Mach. Learn.*, 2003, pp. 912–919.
[186] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, and G. Li, "Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1237–1246.
[187] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[188] S. Cao, W. Lu, and Q. Xu, "GraRep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 891–900.
[189] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.
[190] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 15th Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 849–856.
[191] H. Dai, B. Dai, and L. Song, "Discriminative embeddings of latent variable models for structured data," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 2702–2711.
[192] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2015, *arXiv:1511.05493*.
[193] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.
[194] S. Roweis and Z. Ghahramani, "A unifying review of linear Gaussian models," *Neural Comput.*, vol. 11, no. 2, pp. 305–345, Feb. 1999.
[195] J. Shlens, "A tutorial on principal component analysis," 2014, *arXiv:1404.1100*.
[196] Y. Tian, "GADY: Unsupervised anomaly detection on dynamic graphs," in *Proc. Submitted Comput. Sci. Undergradaute Conf.*, 2025.
[197] Y. Liu, K. Ding, H. Liu, and S. Pan, "GOOD-D: On unsupervised graph out-of-distribution detection," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, Feb. 2023, pp. 339–347.
[198] W. Yu, W. Cheng, C. C. Aggarwal, K. Zhang, H. Chen, and W. Wang, "NetWalk: A flexible deep embedding approach for anomaly detection in dynamic networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2672–2681.
[199] J. Höner, S. Nakajima, A. Bauer, K.-R. Müller, and N. Görnitz, "Minimizing trust leaks for robust Sybil detection," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1520–1528.
[200] C. C. Noble and D. J. Cook, "Graph-based anomaly detection," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 631–636.
[201] Z. You, X. Gan, L. Fu, and Z. Wang, "GATAE: Graph attention-based anomaly detection on attributed networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2020, pp. 389–394.
[202] X. Yuan, N. Zhou, S. Yu, H. Huang, Z. Chen, and F. Xia, "Higher-order structure based anomaly detection on attributed networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 2691–2700.
[203] Z. Li, Q. Wu, F. Nie, and J. Yan, "GraphDE: A generative framework for debiased learning and out-of-distribution detection on graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 30277–30290.
[204] S. Yang, B. Liang, A. Liu, L. Gui, X. Yao, and X. Zhang, "Bounded and uniform energy-based out-of-distribution detection for graphs," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024.
[205] D. Wang, R. Qiu, G. Bai, and Z. Huang, "GOLD: Graph out-of-distribution detection via implicit adversarial latent generation," in *Proc. The 13th Int. Conf. Learn. Represent.*, 2025.
[206] X. Du, J. Yu, Z. Chu, L. Jin, and J. Chen, "Graph autoencoder-based unsupervised outlier detection," *Inf. Sci.*, vol. 608, pp. 532–550, Aug. 2022.
[207] G. Zhang, Z. Yang, J. Wu, J. Yang, S. Xue, H. Peng, J. Su, C. Zhou, Q. Z. Sheng, L. Akoglu, and C. Aggarwal, "Dual-discriminative graph neural network for imbalanced graph-level anomaly detection," in *Proc. Adv. Neural Inf. Process. Syst. 35*, 2022, pp. 24144–24157.
[208] X. Xu, K. Ding, C. Chen, and K. Shu, "MetaGAD: Meta representation adaptation for few-shot graph anomaly detection," in *Proc. IEEE 11th Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2024, pp. 1–10.
[209] Y. Abbahaddou, F. D. Malliaros, J. F. Lutzeyer, A. M. Aboussalah, and M. Vazirgiannis, "Graph neural network generalization with Gaussian mixture model based augmentation," 2024, *arXiv:2411.08638*.
[210] B. Preiss, *Data Structures and Algorithms With Object-Oriented Design Patterns in Java*. Hoboken, NJ, USA: Wiley, 2000.
[211] M. Nguyen, T. Le, T. Pham, and T. Quan, "GEO: An approach for out-of-distribution detection in graph neural networks using energy scoring and one-class learning," in *Proc. 13th Conf. Inf. Technol. Its Appl.* Cham, Switzerland: Springer, 2024, pp. 259–270.
[212] R. Chalapathy, E. Toth, and S. Chawla, "Group anomaly detection using deep generative models," in *Proc. Mach. Learn. Knowl. Discovery Databases, Eur. Conf.*, Dublin, Ireland. Cham, Switzerland: Springer, 2018, pp. 173–189.
[213] R. Cao, S. Xue, J. Li, Q. Wang, and Y. Chang, "FANFOLD: Graph normalizing flows-driven asymmetric network for unsupervised graph-level anomaly detection," 2024, *arXiv:2407.00383*.
[214] N. Yang, K. Zeng, Q. Wu, X. Jia, and J. Yan, "Learning substructure invariance for out-of-distribution molecular representations," in *Proc. 36th Adv. Neural Inf. Process. Syst.*, 2022, pp. 12964–12978.
[215] D. M. Tax and R. P. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, pp. 45–66, Jan. 2004.
[216] S. Alam, S. K. Sonbhadra, S. Agarwal, and P. Nagabhushan, "One-class support vector classifiers: A survey," *Knowl.-Based Syst.*, vol. 196, May 2020, Art. no. 105754.
[217] F. Keller, E. Müller, and K. Bohm, "HiCS: High contrast subspaces for density-based outlier ranking," in *Proc. IEEE 28th Int. Conf. Data Eng.*, Apr. 2012, pp. 1037–1048.
[218] I. Dhillon, Y. Koren, R. Ghani, T. Senator, P. Bradley, R. Parekh, J. He, R. Grossman, and R. Uthurusamy, in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013.
[219] F. Azmandian, A. Yilmazer, J. G. Dy, J. A. Aslam, and D. R. Kaeli, "GPU-accelerated feature selection for outlier detection using the local kernel density ratio," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 51–60.
[220] G. Pang, L. Cao, and L. Chen, "Outlier detection in complex categorical data by modelling the feature value couplings," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1902–1908.
[221] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

MAO NGUYEN received the bachelor's degree in computer science from Ho Chi Minh City University of Science, in 2011. He is currently pursuing the Ph.D. degree in computer science with Ho Chi Minh City University of Technology (HCMUT), Vietnam. He has a strong background in mathematics and theoretical foundations, which informs his research approach in machine learning and intelligent systems. His research interests include principled modeling, deep learning, and analytical perspectives on learning algorithms. He is particularly interested in understanding out-of-distribution generalization from a conceptual and theoretical standpoint.

THIEN PHAM received the Ph.D. degree in computer science from Ho Chi Minh City University of Technology (HCMUT), in 2025. He is currently a Lecturer with the Faculty of Information Technology, Ho Chi Minh City University of Foreign Languages and Information Technology (HUFLIT), Vietnam. His research focuses on machine learning and deep learning methods for time-series forecasting and data-driven modeling. He also works on artificial intelligence applications in semiconductor-related systems, natural language processing, and computer vision. His broader interests include bridging theoretical insights and real-world AI systems.

LONG S. T. NGUYEN (Student Member, IEEE) is currently a Research Assistant with the URA Research Group, Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Vietnam, and an AI Researcher with the Center for Artificial Intelligence Research (CAIR), VinUniversity, Vietnam. His research interests include information retrieval, language models, and question answering systems. He focuses on retrieval-augmented reasoning, efficient inference, and robustness in large language models. His work emphasizes building trustworthy and interpretable intelligent systems for practical deployment. He is particularly interested in domain-specific applications, such as education and law.

DANG LE is currently pursuing the master's degree with the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Japan. His research centers on large language models and their reliability in real-world settings. He is particularly interested in hallucination detection and uncertainty quantification. His work also explores temporary misalignment phenomena and model behavior under distributional shifts. Through this line of research, he aims to improve the robustness and trustworthiness of modern AI systems.

TRUNG M. PHAM is currently pursuing the bachelor's degree in computer science with Ho Chi Minh City University of Technology (HCMUT), Vietnam. He is a Research Assistant with the URA Research Group. His research interests include machine learning, deep learning, and artificial intelligence alignment. He works on applying AI techniques to healthcare-oriented problems. In particular, he is interested in intelligent systems based on biomedical signals, such as EEG. His long-term goal is to develop reliable and human-centered AI for healthcare applications.

HIEU M. PHAM is currently pursuing the bachelor's degree in computer science with Ho Chi Minh City University of Technology (HCMUT), Vietnam. He is a Research Assistant with the URA Research Group. His research interests lie in multimodal healthcare artificial intelligence. He works on integrating signal processing techniques with machine learning models. He is particularly interested in brain–computer interfaces and physiological data analysis. His research aims to improve robustness and interpretability in healthcare AI systems.

DUC Q. NGUYEN received the B.Eng. degree in computer science from Ho Chi Minh City University of Technology (HCMUT), Vietnam, in 2022. He is currently pursuing the Ph.D. degree with the National University of Singapore (NUS), Singapore. His research interests include artificial intelligence, graph representation learning, and computational biology. His work focuses on leveraging structured representations for robust and scalable learning, with a particular emphasis on integrating graph-based modeling with data-driven AI methods.

XINH LE is currently pursuing the Ph.D. degree in computer science with Ho Chi Minh City University of Technology (HCMUT), Vietnam. Her research interests include machine learning and artificial intelligence. She works on natural language processing and representation learning. Her research emphasizes robustness and generalization in data-driven models. She is particularly interested in applying learning-based methods to complex real-world scenarios.

ANH H. HUYNH is currently pursuing the bachelor's degree in computer science with Ho Chi Minh City University of Technology (HCMUT), Vietnam. His research interests include artificial intelligence, deep learning, and graph neural networks. He works on applying AI techniques to healthcare-oriented problems. In particular, he is interested in intelligent systems based on medical imaging analysis and leveraging multimodal architectures. His long-term goal is to develop robust and reliable AI solutions for advanced medical diagnosis.

TRIET T. M. LE is currently a Founding Engineer with Growtrics.ai. He focuses on AI engineering pipelines, microservice architecture, DevOps, and cloud infrastructure. His professional expertise includes designing and implementing cloud-native platforms tailored for scaling AI-generated educational content. His research and development interests involve building robust systems architectures that integrate advanced AI capabilities with large-scale deployment environments.

THO T. QUAN received the B.Eng. degree in information technology from Ho Chi Minh City University of Technology (HCMUT), Vietnam, in 1998, and the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2006. He is currently an Associate Professor with the Faculty of Computer Science and Engineering, HCMUT. He is also the Dean of the Faculty of Computer Science and Engineering, HCMUT. His research interests include artificial intelligence, natural language processing, intelligent systems, and formal methods. He has contributed to both theoretical and applied aspects of AI research.