AutoAnoEval: Semantic-Aware Model Selection via Tree-Guided LLM Reasoning for Tabular Anomaly Detection

Suhee Yoon Sanghyu Yoon Ye Seul Sim Seungdong Yoa
Dongmin Kim Soonyoung Lee Hankook Lee[†*] Woohyung Lim[*]
LG AI Research, [†]Sungkyunkwan University
{suhee.yoon, sanghyu.yoon, ysl.sim, seungdong.yoa
dmkim, soonyoung.lee, w.lim}@lgresearch.ai hankook.lee@skku.edu

Abstract

In the tabular domain, which is the predominant data format in real-world applications, anomalies are extremely rare or difficult to collect, as their identification often requires domain expertise. Consequently, evaluating tabular anomaly detection models is challenging, since anomalies may be absent even in evaluation sets. To tackle this challenge, prior works have generated synthetic anomaly generation rely on statistical patterns, they often overlook domain semantics and struggle to reflect the complex, domain-specific nature of real-world anomalies. We propose AutoAnoEval, a novel evaluation framework for tabular AD that constructs pseudo-evaluation sets with semantically grounded synthetic anomalies. Our approach leverages an iterative interaction between a Large Language Model (LLM) and a decision tree (DT): the LLM generates realistic anomaly conditions based on contextual semantics, while the DT provides structural guidance by capturing feature interactions inherent in the tabular data. This iterative loop ensures the generation of diverse anomaly conditions, ranging from easily detectable outliers to subtle cases near the decision boundary. Extensive experiments on 20 tabular AD benchmarks demonstrate that AutoAnoEval achieves superior model selection performance, with high ranking alignment and minimal performance gaps compared to evaluations on anomalies encountered in practical applications.

1 Introduction

Tabular anomaly detection (Tabular AD) is a fundamental task that identifies unexpected patterns in structured tabular data, with broad applications across finance (Carcillo et al., 2021; Schreyer et al., 2019), cybersecurity (Xu et al., 2018; Brown et al., 2018), healthcare (Choi et al., 2016; Purushotham et al., 2018), and manufacturing (Malhotra et al., 2016; Kharitonov et al., 2022). For these applications, there have been proposed numerous AD models, ranging from classical machine learning to modern deep learning models. Despite these advances, no single tabular AD model has shown consistent superiority across all datasets (Han et al., 2022) due to the inherent heterogeneity of tabular data—such as varying dimensionalities, mixed feature types (*e.g.,* categorical, numerical, textual), and domain-specific anomaly definitions that require semantic understanding. Therefore, evaluating tabular AD models for each given dataset becomes crucial for robust anomaly detection (Zhao et al., 2021; Ding et al., 2024).

Evaluating AD models requires an evaluation set containing both ground-truth normal and abnormal samples; however, constructing an appropriate evaluation set for AD is inherently challenging. In real-world scenarios, anomalies are extremely rare and cost expensive to collect, often resulting in incomplete evaluation sets containing only normal samples. The absence of anomalies prevents reliable model selection and limits the practical deployment of AD models. This raises a fundamental research question: *How can we evaluate tabular AD models in the absence of anomalies, even within the evaluation set?*

To address this challenge, we focus on constructing pseudo evaluation sets with synthetic anomalies, exploiting the recent advances in large language model (LLM) for understanding tabular semantics (Han et al., 2024; Nam et al., 2024; Tsai et al., 2025). While ADBench (Han et al., 2022) explored four categories of synthetic anomalies (*e.g.,* local, global, cluster, dependency) for model selection, their practical performance remains limited, as illustrated in Figure 1. The key observation underlying these discrepancies is that such categories rely solely on numerical distributions, overlooking the rich semantic context of tabular data, where contextual relationships and domain-specific con-

[*] Corresponding authors

Findings of the Association for Computational Linguistics: EACL 2026, pages 3546–3560
March 24-29, 2026

straints fundamentally define anomaly conditions. On the other hand, recent LLM have shown promising capabilities in capturing tabular semantic information, suggesting their potential to generate more realistic anomaly conditions that better reflect real-world complexity.

Motivated by the potential of LLMs to capture domain semantics beyond numerical patterns, in this paper, we introduce **AutoAnoEval**, a novel framework that facilitates the construction of pseudo evaluation sets by synthesizing realistic anomaly conditions. In particular, we propose decision tree (DT)-guided LLM reasoning process, accomplishing two essential properties for reliable best model selection. First, AutoAnoEval ensures **model ranking alignment**, where the relative performance ordering of candidate models in pseudo evaluation matches their true rankings, enabling accurate identification of top-performing models even with subtle performance differences (Shoshan et al., 2023). Second, it achieves **performance gap minimization**, ensuring that pseudo evaluation scores closely approximate real performance, thereby reducing deployment risks from overestimation.

To accomplish the desired properties, our framework first prompts an LLM to generate anomaly conditions that capture domain-specific semantics by leveraging the rich contextual information of each tabular dataset (Section 3.1). To further enrich this process, we introduce an iterative decision tree (DT)-guided LLM reasoning process (Section 3.2). At each iteration, a DT model is trained on the current pseudo evaluation set, and its normal paths are extracted to reveal feature interactions that describe normal behavior. The LLM then extends these paths with semantically inconsistent conditions, resembling realistic failure patterns. Through this process, our framework generates diverse anomaly conditions ranging from obvious to challenging cases near the decision boundary. To ensure the most effective combinations, we further select the best iteration considering condition diversity. Finally, we evaluate candidate AD models on this pseudo evaluation set to identify the best-performing model (Section 3.3). Thereby, our pseudo evaluation sets provide sufficient discriminative power across models while offering reasonable performance estimates that closely approximate those on real-world anomalies.

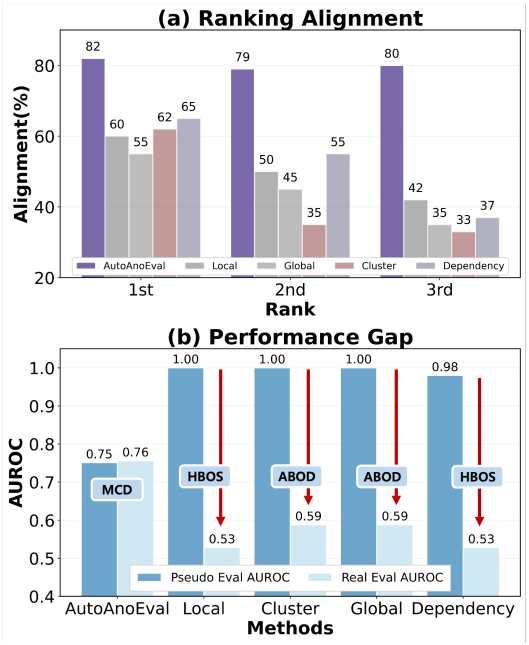

Figure 1: **Comparison of pseudo- vs. real-evaluation performance on gallstone dataset.** (a) Average ranking alignment to ground-truth ranks across diverse seeds. AutoAnoEval consistently achieves high alignment across all ranks, while baseline methods show declining accuracy. (b) Performance gap between pseudo- and real-evaluation results of the best models selected by each method. Baseline methods exhibit significant performance degradation on real evaluation, whereas AutoAnoEval achieves comparable performance with minimal gap.

To sum up, our contributions are summarized as follows:

- We propose **AutoAnoEval**, the first framework that enables rigorous comparison of tabular AD models in the absence of anomalies, even within the evaluation set.

- By constructing pseudo evaluation sets with anomaly conditions that reflect real-world scenarios, **AutoAnoEval** effectively selects the best model. To achieve this, we propose a DT-guided LLM reasoning process that leverages both structural and semantic information of tabular datasets.

- We empirically validate our approach on 20 tabular AD benchmarks, demonstrating its superior best model selection performance by ensuring model ranking alignment and performance gap minimization.

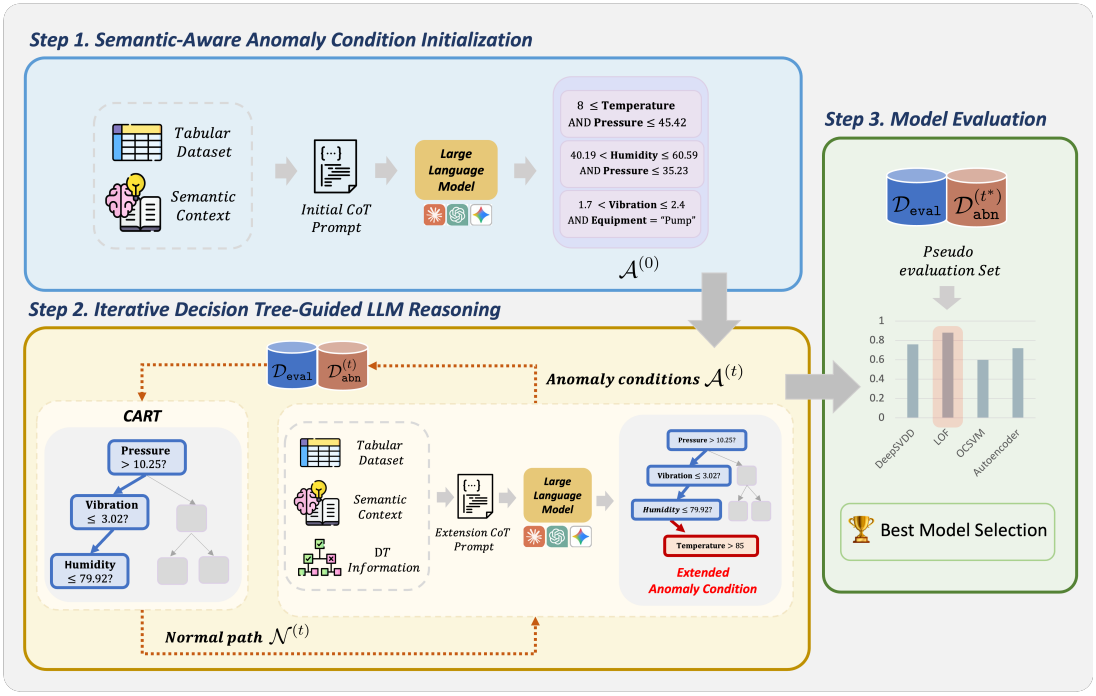

Figure 2: **Overview of AutoAnoEval**. Step 1: LLM generates initial anomaly conditions using tabular data and semantic context. Step 2: Through iterative DT-guided reasoning, CART extracts normal paths which LLM extends into refined anomaly conditions, creating diverse synthetic anomalies from obvious to boundary cases. Step 3: Candidate models are evaluated on the pseudo evaluation set for reliable best model selection.

2 Related Work

2.1 Automatic Model Selection

Selecting the best model in tabular AD remains challenging, especially when anomalies are unavailable during both training and evaluation. Prior work addresses this through meta-learning on historical datasets (Zhao et al., 2021, 2022; Ding et al., 2024), but often struggles to generalize to unseen domains. More recent zero-shot methods (Chen et al., 2025; Yang et al., 2025) utilize LLM prompted with dataset-level summaries, yet still lack domain-specific information. Complementing these efforts, recent advances have demonstrated the potential of synthetic data for model evaluation (Shoshan et al., 2023; van Breugel et al., 2023; Boyeau et al., 2024), showing its effectiveness in estimating performance and enabling fine-grained testing across diverse domains. Synthetic evaluation strategies such as Han et al. (2022) further show that generating predefined anomaly types can guide model selection. However, these methods assume prior knowledge of anomaly characteristics—which is rarely available in practice—and often fail to capture the heterogeneity of real-world patterns. In this work, we aim to construct domain-aware pseudo evaluation sets by synthesizing anomalies from diverse and semantically grounded conditions that reflect real-world scenarios.

2.2 Large Language Models for Tabular Learning

Recent advances have demonstrated the potential of LLM in tabular learning. Several works have explored adapting LLM for tabular prediction via text serialization—Dinh et al. (2022) and Hegselmann et al. (2023) fine-tune GPT-3 and T0, respectively, by converting structured data into natural language formats. Extending this to AD, Tsai et al. (2025) pre-trained LLM with serialized tabular inputs and assign anomaly scores based on the negative log-likelihood. More recently, (Yoon et al., 2025) developed benchmarks with rich metadata and demonstrated zero-shot anomaly detection, validating that LLM can leverage contextual information without task-specific training. To enhance LLM' understanding of tabular structures, Nam et al. (2024) incorporate DT feedback for context-aware feature generation, improving classification performance. Inspired by this DT-LLM synergy, our framework employs iterative DT-guided LLM reasoning specifically for generating realistic anomaly conditions that capture domain-relevant patterns.

3 Methodology

In this section, we introduce **AutoAnoEval**, a novel framework for constructing effective pseudo evalu-

ation sets by synthesizing semantically grounded anomaly conditions. In a nutshell, our approach leverages iterative interaction between LLM and DT to progressively refine conditions toward realistic anomalous patterns. Concretely, we begin by generating initial semantic-aware anomaly conditions using the LLM's understanding of tabular semantics (Section 3.1). We then iteratively enhance these conditions by extending normal paths from DT into semantically inconsistent patterns (Section 3.2). Finally, we evaluate candidate models on the pseudo evaluation set including the refined anomaly conditions and select the best-performing model (Section 3.3). Figure 2 provides an overview of the entire pipeline.

Problem Setup. We consider a tabular data space $\mathcal{X}$ with S columns, expressed as $\mathcal{X} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_S$ where $\mathcal{F}_i$ denotes the i-th column space. Each has a heterogeneous type such as numerical, categorical, or textual. Following the standard one-class setting, the training dataset $\mathcal{D}_{\texttt{train}} \subseteq \mathcal{X}$ consists only of normal samples, and a pool of candidate AD models $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$ is trained on this normal-only data. While standard evaluation assumes access to both normal and anomalous samples, we consider a more realistic scenario where the evaluation set $\mathcal{D}_{\texttt{eval}} \subseteq \mathcal{X}$ also contains only normal samples, making model selection challengingdue to the absence of real anomalies. To address this, we generate a synthetic anomaly set $\mathcal{D}_{\texttt{abn}}$ to construct a pseudo evaluation set, $\widetilde{\mathcal{D}}_{\texttt{eval}} := \mathcal{D}_{\texttt{eval}} \cup \mathcal{D}_{\texttt{abn}}$. Based on the pseudo evaluation set, we select the best model:

$$M^* = \arg\max_{M \in \mathcal{M}} \texttt{Score}(M, \widetilde{\mathcal{D}}_{\texttt{eval}}), \qquad (1)$$

where $\texttt{Score}(\cdot)$ denotes a standard evaluation metric (e.g., AUROC or AUPRC). Our goal is to ensure that the selected model M^* is expected to maintain robust generalization to ensure high performance in actual deployment environments where real anomalies emerge.

3.1 Semantic-Aware Anomaly Condition Initialization

AutoAnoEval begins by generating semantic-aware anomaly conditions through reasoning about intricate domain-specific irregularities potentially encountered in real-world. To this end, we prompt an LLM with an initial prompt $\mathcal{P}_{\texttt{init}}$, which is constructed using two sources of information: a set of randomly selected n normal samples $\mathcal{D}_{\texttt{sub}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ from $\mathcal{D}_{\texttt{eval}}$ and a rich semantic context $\mathcal{C}$ consists of dataset-level descriptions $\mathcal{C}_{\texttt{dataset}}$, feature-level metadata $\mathcal{C}_{\texttt{feat}}$ (*e.g.,* names, units, definition), and statistical summaries $\mathcal{C}_{\text{stat}}$ (*e.g.,* ranges, means, standard deviations, quantiles) of $\mathcal{X}$. To fully leverage the semantic context $\mathcal{C}$, the initial prompt $\mathcal{P}_{\texttt{init}}$ is designed to induce a step-wise chain-of-thought (CoT) reasoning process. Specifically, $\mathcal{P}_{\texttt{init}}$ instructs the LLM to first identify normal patterns of feature relationships and valid value combinations, then generate anomaly conditions that deviate from these semantic patterns.

$$\mathcal{A}^{(0)} = \texttt{LLM}(\mathcal{P}_{\texttt{init}}(\mathcal{D}_{\texttt{sub}}, \mathcal{C})). \qquad (2)$$

As shown in Figure 2, the output $\mathcal{A}^{(0)}$ contains k initial conditions, $a_1^{(0)}, a_2^{(0)}, \ldots, a_k^{(0)}$, where each condition consists of individually valid feature values but semantically inconsistent combination. Based on these conditions, we generate synthetic anomalies by directly sampling data points from regions that satisfy these logical conditions using rule-based generation. The resulting pseudo evaluation set is denoted as $\mathcal{D}_{\texttt{abn}}^{(0)} = \texttt{Sampling}(\mathcal{A}^{(0)})$. While this initial set captures semantic inconsistencies, it often lacks coverage of sufficiently diverse or challenging anomaly scenarios, limiting its ability to reveal differences in model capability.

3.2 Iterative Decision Tree-Guided LLM Reasoning

We now introduce an iterative refinement loop in which DT-based structural learning progressively guides LLM reasoning toward generating refined anomaly conditions.

3.2.1 Normal Path Extraction from Decision Trees

At each iteration $t \in \{1, \ldots, T\}$, we train a CART, $\mathcal{T}^{(t)}$, on the pseudo evaluation set from the previous iteration:

$$\mathcal{T}^{(t)} = \texttt{CART}(\mathcal{D}_{\texttt{eval}} \cup \mathcal{D}_{\texttt{abn}}^{(t-1)}).$$

Intuitively, decision tree paths provide valuable insights from the entire dataset, explicitly revealing which features and thresholds are most effective at distinguishing anomalies from normal patterns. From $\mathcal{T}^{(t)}$, we randomly extract k normal paths that lead to normal leaf nodes:

$$\mathcal{N}^{(t)} = \{n_1^{(t)}, n_2^{(t)}, \ldots, n_k^{(t)}\}.$$

Each $n_i^{(t)}$ is naturally expressed using a simple `if-else` syntax, allowing the LLM to better interpret the linguistic structure (Nam et al., 2024). By extracting paths of varying depths, we capture diverse normal patterns ranging from simple cases (shallow paths) to complex boundary regions (deep paths). This hierarchical structure provides comprehensive guidance for subsequent anomaly generation. To provide additional guidance to the LLM, we compute the conditional entropy of remaining unused features within each path's leaf node, denoted as $\mathcal{H}^{(t)}$, where higher entropy indicates greater value diversity and potential informativeness.

3.2.2 LLM-Based Path Extension to Anomaly Conditions

Using the extracted normal paths $\mathcal{N}^{(t)}$ and the entropy information of unused features $\mathcal{H}^{(t)}$, we guide the LLM to synthesize refined anomaly conditions as follows:

$$\mathcal{A}^{(t)} = \texttt{LLM}(\mathcal{P}_{\texttt{ext}}(\mathcal{D}_{\texttt{sub}}, \mathcal{C}, \mathcal{N}^{(t)}, \mathcal{H}^{(t)})). \quad (3)$$

Specifically, for each normal path $n_i^{(t)}$, our prompt $\mathcal{P}_{\texttt{ext}}$ guides the LLM to construct an anomalous condition $a_i^{(t)} \in \mathcal{A}^{(t)}$ by appending a semantically-violating constraint that contradicts $n_i^{(t)}$. As illustrated in Figure 3, the LLM follows a structured CoT process: it analyzes the normal pattern, identifies informative unused features using entropy, recognizes contextual edge cases, and finally selects anomalous feature-threshold combinations. Each step produces intermediate reasoning, and an explanation for why the generated condition constitutes a realistic yet semantically inconsistent anomaly. Moreover, leveraging the diverse depth of normal paths allows stratified generation from obvious anomalies to challenging cases near the decision boundary. This procedure is iterated for a fixed number and at each iteration t, we generate the synthetic anomaly set as

$$\mathcal{D}_{\texttt{abn}}^{(t)} = \texttt{Sampling}(\mathcal{A}^{(0)} \cup \mathcal{A}^{(1)} \cup \cdots \cup \mathcal{A}^{(t)}),$$

where anomaly conditions from all previous iterations are accumulated. Through this approach, we ensures comprehensive coverage of the anomaly space and prevents the model from relying solely on the most recent refinement.

3.3 Model Evaluation and Selection

To ensure that the most realistic and diverse pseudo evaluation set is used for model comparison, we

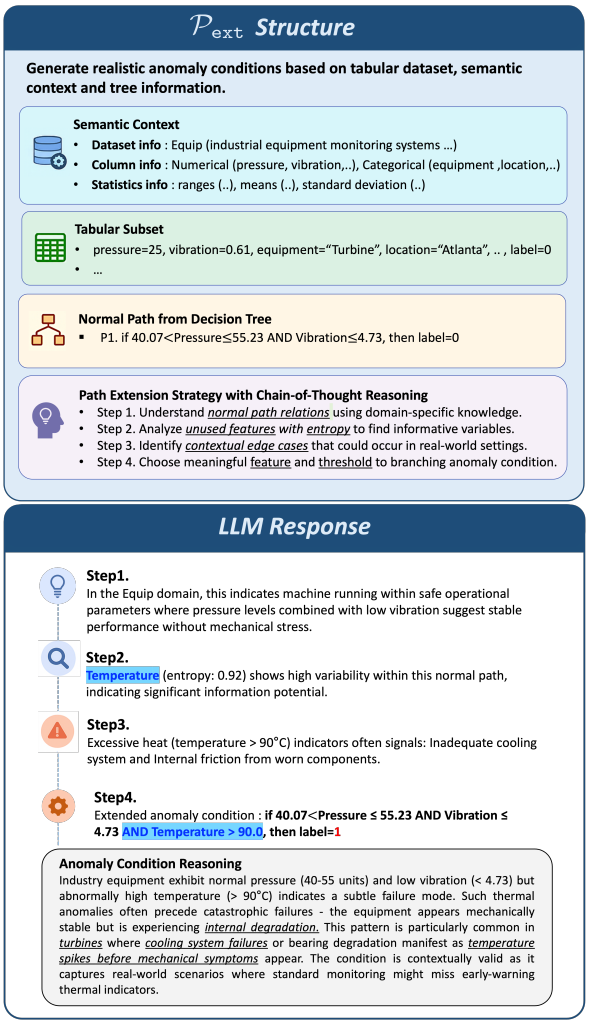

Figure 3: $\mathcal{P}_{\texttt{ext}}$ **structure and LLM response example on *Equip* dataset.**The upper part shows how semantic context, tabular subset, and decision tree paths are integrated into the prompt, while the lower part presents the step-by-step responses and reasoning results about extends normal paths into meaningful anomaly conditions.

determine the best iteration using the anomaly condition diversity score (ACDS):

$$t^* = \arg\max_{t \in \{1,\ldots,T\}} \texttt{ACDS}(t),$$
$$\texttt{ACDS}(t) = \Phi_{\texttt{spread}}^{(t)} + \Phi_{\texttt{sem}}^{(t)}. \quad (4)$$

Specifically, we combine (*i*) a feature spread score $\Phi_{\texttt{spread}}^{(t)}$, which measures how broadly the generated anomalies deviate from normals by averaging the ratio of anomaly range to normal range across features, and (*ii*) a semantic diversity score $\Phi_{\texttt{sem}}^{(t)}$, which captures variety by embedding anomaly condition texts with TF-IDF, computing pairwise cosine similarities, and taking one minus the average similarity. After identifying the best accumulated iteration t^*, we obtain the final pseudo evaluation set $\widetilde{\mathcal{D}}_{\texttt{eval}}$, which combines the normal evaluation

set $\mathcal{D}_{\texttt{eval}}$ with the generated anomaly set $\mathcal{D}_{\texttt{abn}}^{(t^*)}$. We then rigorously evaluate all candidate models $\mathcal{M} = \{M_1, \ldots, M_m\}$ on $\widetilde{\mathcal{D}}_{\texttt{eval}}$. For each model M_i, we compute $\texttt{Score}(M_i, \widetilde{\mathcal{D}}_{\texttt{eval}})$ using a standard evaluation metric (*e.g.,* AUROC or AUPRC) and select the best-performing model, M^*, according to Equation 1.

4 Experiments

In this section, we address a key challenge in tabular AD: *How can we evaluate tabular AD models in the absence of anomalies, even within the evaluation set?* To comprehensively answer this question, we investigate the following research questions:

- **RQ1:** Does AutoAnoEval outperform at best model selection in tabular AD?
- **RQ2:** Does AutoAnoEval improve the model rankings alignment with real evaluation?
- **RQ3:** Does AutoAnoEval reduce the performance gap between pseudo and real evaluation?

4.1 Experiments Settings

Datasets. We conduct our evaluation with 20 benchmarks on ReTabAD (Yoon et al., 2025), which provide tabular AD benchmark including rich textual semantics. Each dataset includes comprehensive descriptions at both the dataset and column levels, encompassing diverse domains such as manufacturing, healthcare, finance, and telecommunications. In our experiments, we follow a strict protocol: the training dataset $\mathcal{D}_{\texttt{train}}$ and the evaluation set $\mathcal{D}_{\texttt{eval}}$ consists solely normal samples. The test set $\mathcal{D}_{\texttt{test}}$ containing real anomalies is strictly reserved for final evaluation—never accessed during training or model selection.

Baselines. Our evaluation encompasses three distinct approaches to the tabular AD model selection:
(a) No model selection. This approach deploys single detectors without selection—reflecting common practice when evaluation set is unavailable.
(b) Model selection without pseudo evaluation. These methods select models using dataset statistics or meta-features. *MetaOD* trains a meta-learner on historical task similarity. *PyOD2* prompts LLMs with statistical summaries and model characteristics. *AD-Agent* uses simple metadata for LLM-based selection.
(c) Model selection with statistical pseudo evaluation. These methods constructs pseudo evaluation sets using synthetic anomalies—*local, global, dependency,* and *cluster*—derived from statistical properties *AD-Bench*. We additionally evaluate a *mixed* strategy combining all four types to better reflect real-world heterogeneity.

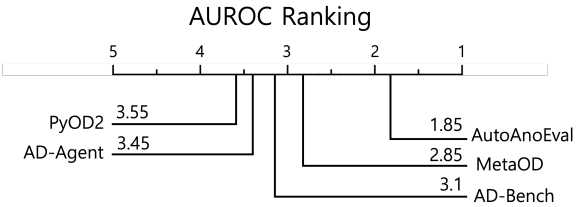

Figure 4: **Average AUROC ranking of selected best models.** Evaluated across 20 tabular AD datasets, AutoAnoEval achieves the lowest (best) average rank of 2.25, substantially outperforming all baseline methods.

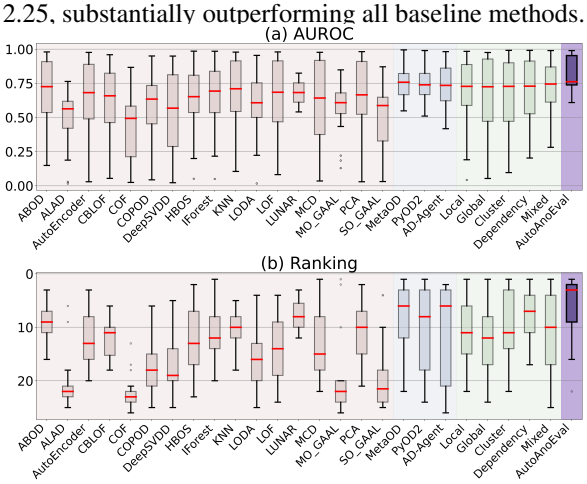

Figure 5: **Comprehensive ranking and AUROC performance across all baselines.** Boxplots over 20 datasets: Color groups indicate different approaches: pink (no model selection), blue (model selection without pseudo-evaluation), green (model selection with statistical pseudo-evaluation), purple (AutoAnoEval). AutoAnoEval achieves the best performance with low variance.

Evaluation Metrics. We evaluate model selection performance by ranking best models based on their Area Under the Receiver Operating Characteristic Curve (AUROC), where a lower rank indicates better performance (rank 1 = best). To assess ranking consistency, we compute the Spearman rank correlation between the model rankings obtained from the pseudo-evaluation set and the test set. Finally, to measure the accuracy of performance estimation, we report the Mean Absolute Error (MAE) between AUROC values. To ensure robustness, each experiment is repeated five times with different random seeds, and the reported scores are averaged across runs.
Implementation Details. We evaluate 18 anomaly detection models spanning classical and deep learning approaches (full list in Appendix 5).

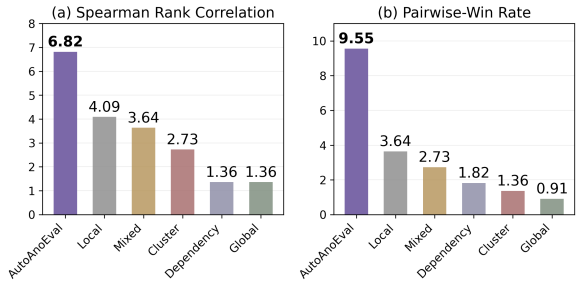

Figure 6: **Win-rate of ranking preservation performance across 20 benchmarks.** (a) Spearman rank correlation between pseudo- and real-evaluation rankings. (b) Pairwise win-rate showing how often relative model orderings are correctly preserved. AutoAnoEval achieves the highest scores in both metrics, demonstrating strong and consistent ranking fidelity.

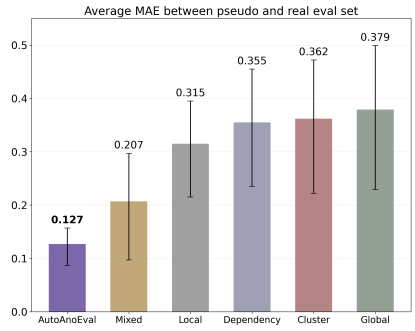

Figure 7: **Average performance gap of selected best models.** AutoAnoEval achieves the lowest MAE, substantially reducing the discrepancy between pseudo- and real-evaluation sets compared to all baselines.

For LLM-based anomaly generation, Gemini-1.5-Flash employed. We generate synthetic anomalies equal to the number of normal samples in the evaluation set, ensuring balanced pseudo evaluation sets. The iterative refinement process runs for $T = 5$ iterations, with $k = 10$ normal paths extracted per iteration.

4.2 Main Results

RQ1: Does AutoAnoEval outperform at best model selection in tabular AD? AutoAnoEval consistently achieves superior performance in best model selection compared to all baselines. Figure 4 presents the average AUROC rankings on the test set $\mathcal{D}_{\texttt{test}}$ for the best models identified by each selection method. Our approach achieves the lowest (*i.e.,* best) average ranking of 1.85, substantially outperforming all baseline methods. Moreover, the boxplot comparison in Figure 5 illustrates both the overall rankings and AUROC distributions across all datasets, including the no model selection setting. AutoAnoEval not only achieves the highest average AUROC and the lowest average rank, but also exhibits markedly lower variance (*i.e.,* shorter whiskers) than other methods. These results indicate that AutoAnoEval enables more stable and reliable model selection, maintaining consistent effectiveness across diverse tabular AD scenarios.

RQ2: Does AutoAnoEval improve the model rankings alignment with real evaluation? AutoAnoEval demonstrates the strongest ranking alignment with real test evaluations, achieving the highest Spearman rank correlation and pairwise win-rate among all methods, as shown in Figure 6. These results are computed across 20 benchmark datasets with 5 random seeds each, and the win-rate represents how often each method wins others in ranking preservation. In detail, the Spearman rank correlation (Figure 6a) quantifies the overall consistency between pseudo-evaluation and real-evaluation rankings, while the pairwise win-rate (Figure 6b) measures how reliably each method maintains the relative orderings between model pairs. AutoAnoEval significantly surpasses all baselines in both metrics, with a clear margin over the second-best method. This robust rank preservation is particularly valuable in real-world deployment scenarios: even when the top-performing model cannot be adopted due to practical constraints, practitioners can still confidently select from among the top-ranked alternatives within a reliable ranking hierarchy.

RQ3: Does AutoAnoEval accurately approximate each model's real performance? AutoAnoEval significantly reduces the performance gap (MAE) between pseudo and real evaluations of selected best models (Figure 7). This represents a 39% reduction compared to the second-best method, demonstrating AutoAnoEval's ability to generate synthetic anomalies that closely reflect real-world characteristics. Such accurate performance approximation is critical in high-stakes domains, where even small mispredictions can incur substantial costs. By providing reliable pre-deployment performance estimates, AutoAnoEval empowers practitioners to make informed decisions that ensure both operational effectiveness and safety.

4.3 Analysis

Ablation on Semantic Context $\mathcal{C}$. To assess the contribution of each context component, we conduct an ablation study across different combinations of context types. As described in Section 3.1,

Table 1: **Performance Comparison across Different Context Types on the AutoAnoEval.** Adding richer semantic context consistently improves both AUROC and MAE.

Context Information			Metrics	
$\mathcal{C}_{\texttt{stat}}$	$\mathcal{C}_{\texttt{feat}}$	$\mathcal{C}_{\texttt{dataset}}$	Avg. AUROC ↓	Avg. MAE ↓
✓	✗	✗	0.755	0.278
✓	✓	✗	0.789	0.154
✓	✓	✓	**0.809**	**0.127**

Table 2: **LLM Reasoning Quality Evaluation across Iterations.** Comparison of five reasoning metrics and overall score across iterations.

Reasoning Metrics	Iter 1	Iter 2	Iter 3	Iter 4
Realism	0.6425	0.7094	0.7128	0.6763
Specificity	0.6992	0.7494	0.7408	0.7150
Boundary Align.	0.5867	0.6600	0.6625	0.5975
Complexity Bal.	0.7450	0.8256	0.7917	0.7575
Anomaly-ness	0.6100	0.6763	0.6655	0.6373
Overall Score	0.6450	**0.7209**	<u>0.7170</u>	0.6633

our framework utilizes three types of context: statistical summaries $\mathcal{C}_{\texttt{stat}}$, feature-level descriptions $\mathcal{C}_{\texttt{feat}}$, and dataset-level descriptions$\mathcal{C}_{\texttt{domain}}$. Table 1 shows that using only $\mathcal{C}_{\texttt{stat}}$ results in limited performance, indicating that statistical information alone is insufficient. Incorporating feature-level context improves performance significantly and the full context setting achieves the best results. This progression demonstrates that rich semantic information is crucial for generating realistic anomalies that capture intricate anomaly scenarios.

Tree-guided LLM Reasoning Quality Evaluation. To assess the effectiveness of our iterative anomaly condition generation process, we quantitatively evaluate the quality of generated conditions at each iteration using five complementary reasoning metrics: realism, specificity, boundary alignment, complexity balance, and anomaly-ness. We employ Gemini-2.0-Pro as an independent evaluator, a more advanced model than the Gemini-1.5-Flash used for generation, to ensure unbiased assessment. As shown in Table 2, the results reveal that condition quality peaks at iterations 2-3, aligning with our ACDS-based best iteration selection results. These findings provide further empirical support for selecting the best iteration rather than defaulting to the final one, preventing the use of sub-optimal synthetic data and enabling a truly robust model evaluation.

Table 3: **Average AUROC and ranking with training-time exposure of AutoAnoEval generated anomalies.** AutoAnoEval yields the best performance, demonstrating the effectiveness of its synthetic anomalies for detector training.

Method	Metrics	
	Avg. AUROC ↑	Avg. Rank ↓
IForest	0.778	7.55
OCSVM	0.803	5.45
LOF	0.761	7.5
DeepSVDD	0.796	6.45
NeuTraL	0.818	4.35
SLAD	0.817	5.45
DIF	0.742	7.7
MCM	0.825	3.95
DRL	0.801	6.25
AnoLLM	0.769	7.35
AutoAnoEval	**0.842**	**3.0**

Effect of Synthetic Anomalies on Training. We explore whether our synthetic anomalies can enhance detector performance through training-time exposure. To this end, we incorporate the generated anomalies into the training process using a contrastive learning framework, where synthetic outliers serve as negative samples to help detectors learn more discriminative decision boundaries. Table 3 demonstrates that this approach yields consistent improvements. Specifically, AutoAnoEval achieves the highest average AUROC and the lowest average rank with synthetic anomaly exposure, indicating more robust performance across diverse datasets. These gains validate that our generated anomalies enable detectors to learn more comprehensive representations of anomalous behaviors when exposed during training. Detailed per-dataset results across all 20 benchmarks are provided in Appendix D.2.

5 Conclusion

This work presented AutoAnoEval, the novel framework that addresses the challenge of model selection in tabular AD when anomalies are absent in both training and evaluation sets. Our key idea lies in constructing pseudo evaluation sets by synthesizing semantic-aware anomaly conditions through iterative decision tree-guided LLM reasoning. This approach effectively captures the complexity of real-world anomalous patterns. Through iterative refinement, AutoAnoEval progressively enriches anomaly conditions from easily detectable cases to subtle boundary scenarios, thereby enabling robust model selection. Experiments on

20 benchmarks confirm that pseudo evaluation sets generated by AutoAnoEval preserve both model ranking alignment and performance fidelity compared to real anomaly evaluation. These results demonstrate that pseudo evaluation can serve as a reliable alternative when real anomalies are unavailable, supporting robust deployment of anomaly detection systems in practical settings.

Limitations. While our framework demonstrates strong performance across diverse tabular benchmarks, it assumes the availability of meaningful semantic context. In domains with limited metadata or poorly structured features, the quality of LLM-generated anomalies may degrade. Additionally, our iterative refinement process incurs computational costs from repeated LLM queries and decision tree training.

Ethical Considerations. The deployment of anomaly detection systems, even when evaluated using our framework, requires careful ethical consideration. While our pseudo evaluation approach reduces the barrier to AD system deployment, practitioners should remain aware that synthetic evaluation cannot capture all real-world complexities and biases. Additionally, We recommend that practitioners should establish monitoring systems to track actual performance post-deployment and compare it with pseudo evaluation estimates to ensure responsible use.

References

Charu C. Aggarwal. 2015. Outlier analysis. In *Data Mining*, pages 237–263. Springer, Cham.

Pierre Boyeau, Anastasios N Angelopoulos, Nir Yosef, Jitendra Malik, and Michael I Jordan. 2024. Autoeval done right: Using synthetic data for model evaluation. *arXiv preprint arXiv:2403.07008*.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. Lof: identifying density-based local outliers. *ACM SIGMOD Record*, 29(2):93–104.

Andy Brown, Aaron Tuor, Brian Hutchinson, and Nicole Nichols. 2018. Recurrent neural network attention mechanisms for interpretable system log anomaly detection. In *Proceedings of the first workshop on machine learning for computing systems*, pages 1–8.

Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, Yacine Kessaci, Frédéric Oblé, and Gianluca Bontempi. 2021. Combining unsupervised and supervised learning in credit card fraud detection. *Information sciences*, 557:317–331.

Sihan Chen and 1 others. 2025. Pyod 2: A python library for outlier detection with llm-powered model selection. In *Companion Proceedings of the ACM on Web Conference 2025*. ACM.

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism.

Xueying Ding, Yue Zhao, and Leman Akoglu. 2024. Fast unsupervised deep outlier model selection with hypernetworks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM.

Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. 2022. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. In *Advances in Neural Information Processing Systems*, volume 35, pages 11763–11784.

Markus Goldstein and Andreas Dengel. 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. In *KI-2012: Poster and Demo Track*, pages 59–63.

Alex Goodge, Bryan Hooi, See Kiong Ng, and Wee Siong Ng. 2022. Lunar: Unifying local outlier detection methods via graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI.

S. Han, J. Yoon, S. O. Arik, and T. Pfister. 2024. Large language models can automatically engineer features for few-shot tabular learning. *arXiv preprint arXiv:2404.09491*.

Songqiao Han and 1 others. 2022. Adbench: Anomaly detection benchmark. In *Advances in Neural Information Processing Systems*, volume 35, pages 32142–32159.

Johanna Hardin and David M Rocke. 2004. Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Computational Statistics & Data Analysis*, 44(4):625–638.

Zengyou He, Xiaofei Xu, and Shengchun Deng. 2003. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 5549–5581. PMLR.

Andrey Kharitonov, Abdulrahman Nahhas, Matthias Pohl, and Klaus Turowski. 2022. Comparative analysis of machine learning models for anomaly detection in manufacturing. *Procedia Computer Science*, 200:1288–1297.

Hans-Peter Kriegel and Arthur Zimek. 2008. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pages 444–452. ACM.

Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. 2020. Copod: Copula-based outlier detection. In *IEEE International Conference on Data Mining (ICDM)*. IEEE.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *IEEE International Conference on Data Mining*, pages 413–422. IEEE.

Yue Liu, Zheng Li, Chenglin Zhou, Yujia Jiang, Jing Sun, Min Wang, and Xiangnan He. 2019. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*.

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*.

J. Nam, K. Kim, S. Oh, J. Tack, J. Kim, and J. Shin. 2024. Optimized feature generation for tabular data via llms with decision tree reasoning. In *Advances in Neural Information Processing Systems*, volume 37, pages 92352–92380.

Tomáš Pevný. 2016. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304.

Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. 2018. Benchmarking deep learning models on large healthcare datasets. *Journal of biomedical informatics*, 83:112–134.

Suresh Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record*, 29(2):427–438.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib A. Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4393–4402. PMLR.

Marco Schreyer, Timur Sattarov, Christian Schulze, Bernd Reimer, and Damian Borth. 2019. Detection of accounting anomalies in the latent space using adversarial autoencoder neural networks. *arXiv preprint arXiv:1908.00734*.

Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, Matan Fintz, and Gérard Medioni. 2023. Synthetic data for model selection. In *Proceedings of the International Conference on Machine Learning*, pages 31633–31656. PMLR.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and Li Chang. 2003. A novel anomaly detection scheme based on principal component classifier. Technical report, Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL.

Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David Wai-Lok Cheung. 2002. Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 535–548, Berlin, Heidelberg. Springer.

Che-Ping Tsai, Ganyu Teng, Phillip Wallis, and Wei Ding. 2025. AnoLLM: Large language models for tabular anomaly detection. In *The Thirteenth International Conference on Learning Representations*.

Boris van Breugel, Nabeel Seedat, Fergus Imrie, and Mihaela van der Schaar. 2023. Can you rely on your model evaluation? improving model evaluation with synthetic test data. In *Advances in Neural Information Processing Systems*, volume 36, pages 1889–1904.

Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, and 1 others. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*, pages 187–196.

Tiankai Yang and 1 others. 2025. Ad-agent: A multi-agent framework for end-to-end anomaly detection. *arXiv preprint arXiv:2505.12594*.

Sanghyu Yoon, Dongmin Kim, Suhee Yoon, Ye Seul Sim, Seungdong Yoa, Hye-Seung Cho, Soonyoung Lee, Hankook Lee, and Woohyung Lim. 2025. Retabad: A benchmark for restoring semantic context in tabular anomaly detection. *arXiv preprint arXiv:2510.02060*.

Houssam Zenati, Manon Romain, Chuan Sheng Foo, Benjamin Lecouat, and Vijay Chandrasekhar. 2018. Adversarially learned anomaly detection. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 727–736. IEEE.

Yue Zhao, Ryan Rossi, and Leman Akoglu. 2021. Automatic unsupervised outlier model selection. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Yue Zhao, Sean Zhang, and Leman Akoglu. 2022. Toward unsupervised outlier model selection. In *2022 IEEE International Conference on Data Mining (ICDM)*, page in press. IEEE.

A Experimental Setting Details.

A.1 Dataset Details.

We conduct our experiments on 20 tabular datasets from ReTabAD (Yoon et al., 2025), a comprehensive benchmark specifically designed for tabular anomaly detection with rich textual semantics. The benchmark spans diverse domains including healthcare (7 datasets: cardiotocography, cirrhosis, glioma, stroke, gallstone, vertebral, wbc), finance (campaign, credit, churn), cybersecurity (backdoor), manufacturing (automobile, equip), and scientific areas such as biology (yeast), astronomy (quasar), and geophysics (seismic). Table 4 provides detailed statistics for each dataset. Each dataset includes comprehensive textual descriptions at both dataset and column levels, providing domain context, feature semantics, and unit information. This semantic richness enables LLMs to leverage their language understanding capabilities for domain-aware reasoning about anomaly patterns.

A.2 Tabular AD Models.

We evaluate 17 diverse anomaly detection methods spanning both classical machine learning and deep learning approaches (Table 5). The classical methods (11 models) include density-based approaches (LOF, COF, CBLOF), distance-based methods (KNN, ABOD), tree-based isolation techniques (IForest), statistical approaches (PCA, MCD, COPOD, HBOS), and ensemble methods (LODA). The deep learning methods (6 models) employ neural architectures to learn complex representations of normal data. These include reconstruction-based approaches (AutoEncoder), methods combining deep learning with traditional one-class classification (DeepSVDD), self-supervised techniques (LUNAR), and generative adversarial networks (ALAD, MO-GAAL, SO-GAAL) that learn to distinguish normal from anomalous patterns through adversarial training. Table 5 provides detailed descriptions of each method. All methods follow the unsupervised paradigm, training exclusively on normal or predominantly normal dataset.

B Core Algorithm of AutoAnoEval.

Algorithm 1: AutoAnoEval: Tree-Guided LLM Reasoning for Model Selection

Input: Tabular dataset $\mathcal{X}$, Training set $\mathcal{D}_{\text{train}}$, Evaluation set $\mathcal{D}_{\text{eval}}$, Semantic context $\mathcal{C}$, Candidate models $\mathcal{M}$, Iterations T

Output: Best model M^*

`// Phase 1: Semantic-Aware Initialization`

$\mathcal{X}_{\text{sub}} \leftarrow \text{RandomSample}(\mathcal{D}_{\text{eval}}, n)$;

$\mathcal{C}_{\text{stat}} \leftarrow \text{ComputeStatistics}(\mathcal{X})$;

$\mathcal{C} \leftarrow \mathcal{C}_{\text{dataset}} \cup \mathcal{C}_{\text{feat}} \cup \mathcal{C}_{\text{stat}}$;

$\mathcal{A}_0 \leftarrow \text{LLM}(\mathcal{P}_{\text{init}}(\mathcal{X}_{\text{sub}}, \mathcal{C}))$;

$\mathcal{D}_{\text{abn}}^{(0)} \leftarrow \text{Sampling}(\mathcal{A}_0)$;

$\mathcal{A} \leftarrow \mathcal{A}_0$;

`// Phase 2: Iterative Tree-Guided Refinement`

for $t = 1$ **to** T **do**

- $\mathcal{T}_t \leftarrow \text{CART}(\mathcal{D}_{\text{eval}} \cup \mathcal{D}_{\text{abn}}^{(t-1)})$;
- $\mathcal{N}_t \leftarrow \text{ExtractNormalPaths}(\mathcal{T}_t, k)$;
- $\mathcal{H}_t \leftarrow \text{ComputeConditionalEntropy}(\mathcal{T}_t, \mathcal{N}_t)$;
- $\mathcal{A}_t \leftarrow \text{LLM}(\mathcal{P}_{\text{ext}}(\mathcal{X}_{\text{sub}}, \mathcal{C}, \mathcal{N}_t, \mathcal{H}_t))$;
- $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{A}_t$;
- $\mathcal{D}_{\text{abn}}^{(t)} \leftarrow \text{Sampling}(\mathcal{A})$;

`// Phase 3: Model Evaluation and Selection`

$t^* \leftarrow \arg\max_{t \in \{1,\ldots,T\}} \text{ACDS}(t)$;

$\widetilde{\mathcal{D}}_{\text{eval}} \leftarrow \mathcal{D}_{\text{eval}} \cup \mathcal{D}_{\text{abn}}^{(t^*)}$;

foreach $M_i \in \mathcal{M}$ **do**

- $s_i \leftarrow \text{Score}(M_i, \widetilde{\mathcal{D}}_{\text{eval}})$;

$M^* \leftarrow \arg\max_{M_i \in \mathcal{M}} s_i$;

return M^*;

Table 4: **Dataset statistics and domain information.** Summary of the 20 tabular datasets from ReTabAD (Yoon et al., 2025) used in our experiments. Each dataset includes rich textual metadata at both the dataset and column levels, enabling domain-aware semantic reasoning for anomaly condition generation.

Dataset Name	Domain	Datapoints	Columns	Normal Count	Anomaly Count	Anomaly Ratio (%)
automobile	Manufacturing	159	26	117	42	26.42
backdoor	Network	50,000	42	48,780	1,220	2.44
campaign	Finance	7,842	16	6,056	1,786	22.77
cardiotocography	Healthcare	2,126	21	1,655	471	22.15
census	Demographics	50,000	41	47,121	2,879	5.76
churn	Telecommunications	7,032	20	5,163	1,869	26.6
cirrhosis	Healthcare	247	17	165	82	33.20
covertype	Environment	50,000	12	49,520	480	0.96
credit	Finance	30,000	23	23,364	6,636	22.12
equip	Manufacturing	7,672	6	6,905	767	10.00
gallstone	Healthcare	241	38	161	80	33.20
glass	Forensics	214	9	163	51	23.83
glioma	Healthcare	730	23	487	243	33.29
quasar	Astronomy	50,000	8	40,520	9,480	18.96
seismic	Geophysics	2,584	18	2,414	170	6.58
stroke	Healthcare	4,909	10	4,700	209	4.26
vertebral	Healthcare	310	6	210	100	32.26
wbc	Healthcare	535	30	357	178	33.27
wine	Chemistry	178	13	130	48	26.97
yeast	Biology	1,484	8	1,389	95	6.40

Table 5: **Summary of baseline anomaly detection methods.** Classical and deep learning models are grouped separately. All methods are unsupervised, trained only on normal or unlabeled data.

Model	**Description**
IForest (Liu et al., 2008)	Isolation Forest; isolates points via random partitions, anomalies have shorter paths.
KNN (Ramaswamy et al., 2000)	k-Nearest Neighbors; anomalies lie far from dense neighbor clusters.
LOF (Breunig et al., 2000)	Local Outlier Factor; detects points with low local density vs neighbors.
PCA (Shyu et al., 2003)	Principal Component Analysis; uses reconstruction error in low-dimensional space.
COF (Tang et al., 2002)	Connectivity-based Outlier Factor; considers chaining distance for density-robust detection.
COPOD (Li et al., 2020)	Copula-based; estimates tail probabilities without parameters.
HBOS (Goldstein and Dengel, 2012)	Histogram-based; uses inverse histogram density, assumes feature independence.
MCD (Hardin and Rocke, 2004)	Minimum Covariance Determinant; detects outliers via robust Mahalanobis distance.
LODA (Pevný, 2016)	Lightweight Online Detector; ensemble of sparse 1D projections with histograms.
CBLOF (He et al., 2003)	Cluster-Based LOF; combines clustering and local density for anomaly scores.
ABOD (Kriegel and Zimek, 2008)	Angle-Based Outlier Detection; identifies outliers via variance of point angles.
AutoEncoder (Aggarwal, 2015)	Learns compressed representation; high reconstruction error signals anomaly.
DeepSVDD (Ruff et al., 2018)	Maps data into minimum-volume hypersphere; distant points are anomalous.
LUNAR (Goodge et al., 2022)	Self-supervised + GNN; captures normal/abnormal representation without labels.
ALAD (Zenati et al., 2018)	GAN-based bidirectional mapping; misaligned mappings indicate anomalies.
MO-GAAL (Liu et al., 2019)	Multiple GAN generators; produce diverse synthetic outliers for robust detection.
SO-GAAL (Liu et al., 2019)	Single GAN generator; adversarially refines normal–abnormal boundary.

C Prompt Templates and Response Examples

Prompt A. Semantic-Aware Anomaly Condition Prompt

```
Generate realistic but unusual anomaly conditions based on domain knowledge, dataset context, and
data statistics.

## Dataset Context: {dataset_name}
Dataset Description: {description}
Source: {source}

## Feature Descriptions (for each feature)
### Numerical Features:
{feature_name}: {feature_description}

### Categorical Features:
{feature_name}: {feature_description}

## Data Statistics
- Range: [{min:.3f}, {max:.3f}]
- Mean±Std: {mean:.3f}±{std:.3f}
- Quartiles: Q1={q25:.3f}, Median={median:.3f}, Q3={q75:.3f}

## Normal Sample Example
Sample 0: {feature_1 name}={feature_1 value}, ... , {feature_k name}={feature_k value}
...
Sample 10: {feature_1 name}={feature_1 value}, ... , {feature_k name}={feature_k value}

DOMAIN-AWARE Strategy Guidelines:
## Chain-of-Thought Analysis Process:
### Step 1: Understand Normal Patterns Based on the dataset context and sample data above:
- What typical patterns and value ranges exist in normal data?
- Which features and their combinations commonly co-occur in normal samples?

### Step 2: Identify Domain Constraints Using feature descriptions and domain knowledge:
- What relationships should logically exist between features and what combinations are valid?
- What business/physical constraints apply to this domain?

### Step 3: Design Anomaly Conditions For each anomaly, systematically consider:
- How does this violate normal patterns identified in Step 1?
- Why would this be unusual given domain constraints from Step 2?

## Pattern Analysis:
- First understand what makes samples "normal" in this domain and identify realistic deviations from
these patterns
- Feature Combinations: Combine 2-5 semantically related features
- Domain Logic Violations: Create conditions that violate expected relationships


Response Format (JSON only):
{
  "anomaly_conditions":[
    {"condition_id": 1,
        "chain_of_thought_reasoning": {
            "step1_understand_normal_patterns": "Normal samples show {Feature_A} and {Feature_B} positively correlated...",
            "step2_identify_domain_constraints": "{Feature_A} represents... and should logically relate to {Feature_B}...",
            "step3_design_anomaly_condition": "This combination violates the expected relationship because..." },
            "anomaly_condition": "{Feature_A} > {threshold} AND {Feature_B} < {threshold}",
            "contrast_with_normal": "Normal samples typically have both features in similar ranges"
    }
  ]}
```

Response A. Semantic-Aware Anomaly Condition Example

```
{
  "anomaly_conditions": [
    {
      "condition_id": 1,
      "chain_of_thought_reasoning": {
        "step1_understand_normal_patterns": "Typical cirrhosis patients show serum bilirubin < 3 mg/dl with
                                             albumin > 3.5 gm/dl and prothrombin time < 11 sec, indicating
                                             stable liver function.",
        "step2_identify_domain_constraints": "Bilirubin reflects liver excretory function, albumin reflects
                                              synthetic capacity, and prothrombin time indicates clotting
                                              ability. Normally, impaired bilirubin clearance is accompanied by
                                              decreased albumin and prolonged prothrombin time.",
        "step3_design_anomaly_condition": "Extremely high bilirubin with still-normal albumin and prothrombin
                                          time is inconsistent with typical disease progression and may
                                          indicate lab error, isolated cholestasis, or rare subtypes."
      },
      "anomaly_condition": "Bilirubin > 10.0 AND Albumin > 3.5 AND Prothrombin < 11.0",
      "contrast_with_normal": "Normal progression shows bilirubin elevation coupled with low albumin and
                               prolonged prothrombin; this condition breaks that linkage"
    },
    {
      "condition_id": 2,
      "chain_of_thought_reasoning": {
        "step1_understand_normal_patterns": "Patients with advanced cirrhosis (Stage ≥3) usually present with
                                             ascites=Yes, hepatomegaly=Yes, and edema despite diuretic therapy.",
        "step2_identify_domain_constraints": "These clinical signs correlate with portal hypertension and
                                              decompensation; absence of ascites or edema in late stages is rare.",
        "step3_design_anomaly_condition": "Stage 4 cirrhosis with no ascites and no edema contradicts typical
                                          decompensated state and suggests misclassification or atypical
                                          presentation."
      },
      "anomaly_condition": "Stage = 4 AND Ascites = 'No' AND Edema = 'no edema and no diuretic therapy for
                            edema'",
      "contrast_with_normal": "Normal Stage 4 cases show fluid retention signs; this combination indicates
                               atypical or inconsistent record"
    }
  ]
}
```

Figure 8: **Prompt–Response pair for anomaly condition initialization.** (**Prompt A**) guides the LLM to generate initial anomaly conditions based on semantic and statistical context, while (**Response A**) presents representative LLM-generated anomaly conditions following this guidance on Cirrhosis dataset.

Prompt B. Decision Tree-Guided LLM Reasoning Prompt

```
EXTEND Normal Decision Tree Paths by adding ONE MORE SPLIT

Iteration: {iteration + 1}
[Same dataset context, Feature Description, Data Statistics, Normal Sample Example as in Prompt A]

##NORMAL PATHS from Decision Tree (for each normal path)
{normal path} (Depth: {depth}, Samples: {n_samples}):
Current Path: {condition_1} AND {condition_2} AND ... → NORMAL
Confidence: {confidence:3f}

##Feature Diversity Analysis (Top 3 features with highest entropy)
{feature_name}: Conditional Entropy {entropy:3f}

DOMAIN-AWARE Strategy Guidelines:
## Chain-of-Thought Analysis Process:
###Step1: Understand Path & Domain
- Why do these conditions lead to NORMAL samples and what semantic meaning do features reveal?
###Step2: Analyze Features & Patterns:
- Which unused features show high diversity and what combinations would violate expected patterns?
###Step3: Identify Edge Cases
- What boundary conditions create contextual anomalies?
###Step4: Select Constraints
- Choose semantically meaningful feature, operator and threshold creating domain-specific anomalies

Extension Example:
Original: {Feature_A} <= 5.0 AND {Feature_B} > 10.0 → NORMAL

Extended:
└─ {Feature_A} <= 5.0 AND {Feature_B} > 10.0 AND {Feature_C} > threshold→ NORMAL
└─ {Feature_A} <= 5.0 AND {Feature_B} > 10.0 AND {Feature_C} <= threshold → ABNORMAL

Response Format (JSON only):
{extended_paths": [
  {"original_path_id": 1,
    "chain_of_thought_reasoning": {
            "step1_understand_path_domain": "This path represents... and features reveal...",
            "step2_analyze_features_patterns": "Unused {Feature_C} shows entropy 0.85, combinations
                                               with {Feature_C} > threshold would violate...",
            "step3_identify_edge_cases": "Edge occur when {Feature_C} exceeds normal range in this
                                         context...",
            "step4_select_feature_threshold": "Selected {Feature_C} with {threshold} because it
                                              creates meaningful domain anomalies..."
        },
    "original_path_conditions": "{Feature_A} <= 5.0 AND {Feature_B} > 10.0",
    "extended_abnormal_condition": "{Feature_A} <= 5.0 AND {Feature_B} > 10.0
                                    AND {Feature_C} <= threshold",
    "extended_abnormal_condition_reasoning": {explanation of anomaly conditions}
  }
]}
```

Response B. Decision Tree-Guided LLM Reasoning Example

```
{
  "extended_paths": [
    {
      "original_path_id": 1,
      "chain_of_thought_reasoning": {
        "step1_understand_path_domain": "This path represents patients who are not experiencing ascites
                                         (Ascites = No) and have relatively high albumin levels (> 3.5),
                                         suggesting preserved liver synthetic function and no fluid accumulation.
                                         These patients are typically in early-stage liver disease with
                                         favorable prognosis.",
        "step2_analyze_features_patterns": "Unused feature 'Bilirubin' shows entropy 0.88, indicating moderate
                                            variability. In this path, patients with low bilirubin are expected
                                            due to preserved liver function. Introducing high bilirubin levels (>
                                            10.0 mg/dL) contradicts this assumption and suggests acute liver
                                            dysfunction despite other indicators being normal.",
        "step3_identify_edge_cases": "Edge cases arise when bilirubin is abnormally high (> 10.0) even though
                                      albumin remains high and ascites is absent. This inconsistency implies an
                                      underlying hidden pathology such as hemolysis or early cholestasis not
                                      yet reflected in albumin synthesis or fluid retention.",
        "step4_select_constraints": "Selected bilirubin > 10.0 as this represents a clinically significant
                                     threshold where jaundice and liver clearance issues emerge, creating
                                     a realistic yet abnormal clinical contradiction in this otherwise
                                     normal path."
      },
      "original_path_conditions": "Ascites = No AND Albumin > 3.5",
      "extended_abnormal_condition": "Ascites = No AND Albumin > 3.5 AND Bilirubin > 10.0",
      "extended_abnormal_condition_reasoning": "Normal patients without ascites and with high albumin typically
                                                show low bilirubin levels, indicating good liver function. However,
                                                high bilirubin in this context suggests an early-stage dysfunction
                                                masked by preserved synthesis—an edge case that breaks expected
                                                clinical coherence and raises red flags for closer evaluation."

    },
    {
      "original_path_id": 2,
      "chain_of_thought_reasoning": {
        "step1_understand_path_domain": "This path captures patients with high platelet counts (> 250) and no
                                         hepatomegaly (Hepatomegaly = No), indicating absence of significant
                                         liver fibrosis or portal hypertension. These patients are typically
                                         not in the late stages of cirrhosis.",
        "step2_analyze_features_patterns": "Unused feature 'Prothrombin' has entropy 0.91. In this path,
                                            prothrombin time is expected to be normal (< 12.0) due to intact
                                            liver synthetic ability. Elevated prothrombin time (> 14.0) would
                                            contradict this and signal synthetic failure.",
        "step3_identify_edge_cases": "When patients have high platelet count (suggesting low fibrosis) but show
                                      elevated prothrombin time (> 14.0), it indicates discordance between
                                      expected clotting function and actual performance—hinting at hidden liver
                                      dysfunction not visible via platelet count.",
        "step4_select_ constraints": "Selected prothrombin > 14.0 as this threshold marks a critical
                                      deviation from expected clotting ability, serving as an early signal
                                      of decompensation in patients not otherwise flagged by structural
                                      indicators like hepatomegaly."
      },
      "original_path_conditions": "Platelets > 250 AND Hepatomegaly = No",
      "extended_abnormal_condition": "Platelets > 250 AND Hepatomegaly = No AND Prothrombin > 14.0",
      "extended_abnormal_condition_reasoning": "This condition challenges the expected link between structural
                                                liver health and clotting function. Normally, high platelets and
                                                absence of hepatomegaly suggest preserved liver state, but
                                                elevated prothrombin indicates early-stage decompensation—making
                                                this a medically plausible anomaly."

    }
  ]
}
```

Figure 9: **Prompt–Response pair for path extension-based anomaly generation.** (**Prompt B**) guides the LLM to extend decision tree paths using conditional entropy, and (**Response B**) illustrates representative refined anomaly conditions generated through this process on Cirrhosis dataset..

D Full Performance Results.

D.1 Full Best Model Selection Results.

Table 6 reports the full AUROC performance of all baseline methods and our proposed approach on the AutoAnoEval benchmark. The results are obtained by evaluating the best model selected by each method on the ground-truth test set containing both normal and anomalous samples. For AutoEncoder and DeepSVDD, we directly report their single-model performance without model selection, as they are widely used one-class baselines. The other methods—MetaOD, PyOD2, AD-Agent, ADBench (*mixed*), and AutoAnoEval —perform model selection for each dataset. In particular, ADBench uses a mixture of local, global, cluster, and dependency-type synthetic anomalies as validation signals. From the table, we observe that our method, AutoAnoEval, achieves the best average AUROC score and the lowest average rank across all datasets, outperforming existing model selection approaches. This demonstrates the effectiveness of our synthetic anomaly generation strategy, which enables more reliable model selection and ultimately yields stronger anomaly detection performance on real-world test data.

D.2 Full Training-time Anomaly Exposure Results

To evaluate the utility of synthetic anomalies for detector training, we incorporate the generated outliers using a contrastive learning framework. Specifically, we maintain the same ratio of normal samples to synthetic outliers as used in our evaluation protocol. During training, synthetic anomalies serve as negative samples in the contrastive objective, enabling detectors to learn more discriminative representations by exposure to diverse anomaly patterns. Each experiment is conducted with five independent runs using different random seeds to ensure statistical reliability.

Table 7 presents the complete dataset-level results for all 20 datasets and 11 methods. The consistent improvements across diverse datasets validate that our synthetic anomalies capture domain-specific failure modes rather than arbitrary statistical deviations. The synthetic outliers effectively serve as an auxiliary anomaly set, providing detectors with exposure to realistic failure scenarios .

E Comparison of the LLMs

Table 8 summarizes the average AUROC and overall ranking across the evaluated LLMs. Overall, all models demonstrate reasonably strong performance, with average AUROC scores, indicating that large language models can effectively support anomaly detection in our setting. Among them, **Gemini-2.0-pro** achieves the best overall performance, obtaining the highest average AUROC (0.828) and the lowest average rank (1.72). **Gemini-1.5-flash** shows the second-best performance. Given its efficiency-oriented design, this suggests a favorable balance between computational cost and predictive performance.

Table 8: **Average AUROC and ranking comparison across different LLMs used in AutoAnoEval.**

LLM	Metrics	
	Avg. AUROC ↑	Avg. Rank ↓
GPT-4o-mini	0.795	2.55
GPT-4.1	0.792	2.89
Claude-3.7-sonnet	0.755	3.21
Gemini-1.5-flash	0.815	1.85
Gemini-2.0-pro	0.828	1.72

F Comparison of the DTs

Many tabular prediction tasks show that decision trees (DTs) often outperform deep learning models by capturing fine-grained local patterns in structured data. Prior work (Nam et al., 2024) also demonstrates that DTs can effectively transfer structural knowledge to LLMs through interpretable decision paths. Motivated by this, our method leverages DT-derived paths to ground anomaly generation in local feature interactions and decision boundaries. We adopt the CART variant due to its binary splits, which yield short and interpretable if–else paths aligned with LLM reasoning. In contrast, J48 (C4.5) produces more complex multi-way splits. As shown in Table 9, CART consistently achieves higher average AUROC and better rankings.

Table 9: **Average AUROC and ranking comparison across decision tree algorithms used in AutoAnoEval.**

Method	Metrics	
	Avg. AUROC ↑	Avg. Rank ↓
MetaOD	0.768	3.63
PyOD2	0.760	4.29
AD-Agent	0.739	4.62
ADBench	0.760	3.89
AutoAnoEval (J48)	0.798	2.32
AutoAnoEval (CART)	**0.815**	**1.97**

Table 6: **AUROC performance of best-selected models on the AutoAnoEval.** The best results per dataset are in **bold**, and the second-best are underlined. Results are reported on ground-truth test sets containing both normal and anomalous samples, averaged over five independent runs. Our method AutoAnoEval achieves the highest average AUROC score and the best average rank across datasets.

Dataset	Baselines				AutoAnoEval
	MetaOD	PyOD2	AD-Agent	ADBench	
automobile	0.758	0.682	0.618	0.748	**0.765**
backdoor	0.812	0.800	0.862	**0.870**	0.864
campaign	0.728	0.739	0.728	**0.758**	0.738
cardiotocography	0.762	0.794	**0.826**	0.726	0.759
census	0.643	0.652	0.633	0.611	**0.707**
churn	0.558	0.511	0.418	0.613	**0.671**
cirrhosis	0.820	0.823	0.841	**0.867**	0.784
covertype	**0.995**	0.990	0.963	0.990	0.991
credit	0.666	**0.667**	0.622	0.527	0.610
equip	0.983	0.924	0.981	0.970	**0.987**
gallstone	0.573	0.558	0.563	0.526	**0.670**
glass	0.975	0.971	0.980	0.595	**0.984**
glioma	0.652	0.639	0.652	0.690	**0.740**
quasar	0.672	0.700	0.500	**0.974**	0.972
seismic	0.715	0.699	0.735	**0.744**	0.740
stroke	0.739	0.663	0.679	0.661	**0.742**
vertebral	0.549	0.724	0.465	0.582	**0.740**
wbc	**0.956**	0.936	0.890	0.916	0.953
wine	0.982	0.937	0.982	0.980	**0.986**
yeast	0.822	0.789	0.846	0.852	**0.889**
Average AUROC	0.768	0.760	0.739	0.760	**0.815**
Average Rank	2.85	3.55	3.45	3.10	**1.85**

Table 7: **AUROC performance with training-time exposure of AutoAnoEval-generated anomalies.** The best results per dataset are in **bold**, and the second-best are underlined. Results show performance gains when detectors are trained with synthetic outliers produced by AutoAnoEval, averaged over five independent runs.

Dataset	Baselines										AutoAnoEval
	IForest	OCSVM	LOF	DeepSVDD	NeuTraL	SLAD	DIF	MCM	DRL	AnoLLM	
automobile	0.648	0.645	0.688	0.845	0.810	0.780	0.647	0.801	0.762	0.625	**0.887**
backdoor	0.888	0.876	0.702	0.800	0.948	0.935	**0.973**	0.961	0.902	0.885	0.951
campaign	0.732	0.700	0.669	0.721	0.760	0.717	0.677	0.761	0.737	0.738	**0.774**
cardiotocography	0.830	0.868	0.826	0.813	0.791	0.764	0.740	0.843	0.847	0.845	**0.955**
census	0.620	0.647	0.596	0.712	0.686	0.687	0.726	0.729	0.728	0.737	**0.753**
churn	0.506	**0.651**	0.465	0.573	**0.651**	0.571	0.494	0.580	0.553	0.557	0.523
cirrhosis	0.849	0.823	0.843	0.807	0.831	0.810	0.820	0.830	0.814	0.850	**0.922**
covertype	0.968	0.998	0.980	0.971	0.998	0.999	0.998	0.998	0.995	0.990	0.995
credit	0.639	0.717	0.590	0.638	0.690	0.691	0.689	0.671	0.680	0.495	0.652
equip	0.980	0.987	0.987	0.984	0.985	0.981	0.987	0.986	0.984	0.986	**0.995**
gallstone	0.650	0.671	0.704	0.731	0.780	0.758	0.649	0.757	0.745	0.574	**0.819**
glass	0.944	0.959	**0.974**	0.963	0.968	0.952	0.936	0.966	0.949	0.907	0.969
glioma	0.722	0.725	0.711	0.766	0.784	**0.886**	0.686	0.790	0.764	0.708	0.822
quasar	0.915	0.930	**0.966**	0.804	0.954	0.952	0.793	0.950	0.897	0.957	0.963
seismic	0.710	0.710	0.615	0.729	0.716	0.727	0.736	0.716	0.714	**0.746**	0.743
stroke	0.679	0.716	0.625	0.722	0.703	0.678	0.705	0.753	0.727	0.687	**0.828**
vertebral	0.500	0.638	0.540	0.581	**0.697**	0.679	0.340	0.615	0.477	0.597	0.647
wbc	0.961	**0.969**	0.960	**0.969**	0.785	0.938	0.679	0.957	0.954	0.871	0.938
wine	0.987	0.957	0.974	0.974	**0.988**	0.980	0.760	0.979	0.958	0.933	0.958
yeast	0.838	0.875	0.803	0.820	0.841	0.846	0.815	0.856	0.841	**0.882**	**0.882**
Average AUROC	0.778	0.803	0.761	0.796	0.818	0.817	0.742	0.825	0.801	0.769	**0.842**
Average Rank	7.55	5.45	7.5	6.45	4.35	5.45	7.7	3.95	6.25	7.35	**3**