Ano2Rule: Rule-Based Global Interpretation for Unsupervised Anomaly Detection in Security

Ruoyu Li*, Yu Zhang*, Qing Li, *Senior Member, IEEE*, Nengwu Wu, Yong Jiang, *Member, IEEE*, Weizhi Meng, *Senior Member, IEEE*, and Laizhong Cui†, *Senior Member, IEEE*

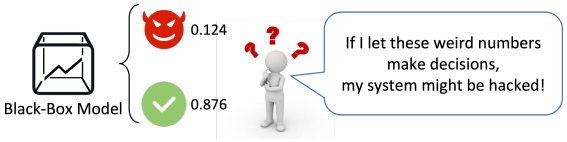
***Abstract*—In the realm of cybersecurity, unsupervised anomaly detection models have emerged as pivotal tools for identifying novel threats in dynamic and evolving environments. However, the opaque nature of these black-box models presents a significant barrier to their adoption in high-stakes applications, where model interpretability is essential for trust and deployment. This paper presents a rule-based approach called Ano2Rule that enhances the interpretability of unsupervised anomaly detection. First, we propose the concept of *distribution decomposition rules* that decompose the complex distribution of normal data into multiple compositional distributions. To find such rules, we design an unsupervised Interior Clustering Tree that incorporates the model prediction into the splitting criteria. Then, we propose the Compositional Boundary Exploration (CBE) algorithm to obtain the *boundary inference rules* that estimate the decision boundary of the original model on each compositional distribution. By merging these two types of rules into a rule set, we can present the inferential process of the unsupervised black-box model in a human-understandable way, and build a surrogate rule-based model for online deployment at the same time. We validate Ano2Rule through extensive experiments on diverse real-world datasets, including network intrusion detection and IoT security, demonstrating superior fidelity and robustness compared to baseline methods. The results show that Ano2Rule achieves high fidelity with the original model's predictions while providing human-understandable insights.**

I. Introduction

In recent years, with the rapid development of machine learning (ML) and deep learning (DL) in the field of anomaly detection, the capability to detect malicious behaviors in security applications has been significantly enhanced. This includes applications such as intrusion detection [1]–[3], malware identification [4], [5], system monitoring [6], [7], and Internet of Things (IoT) network surveillance [8]–[10]. These technologies demonstrate superior performance in terms of accuracy and generalization capabilities. Among these studies, unsupervised anomaly detection exhibits greater potential because it detects potential malicious activities by identifying deviations from normal behavior. Compared to supervised anomaly detection methods, this kind of approach is particularly suitable for detecting unknown attacks and novel threats, offering the following advantages in network security applications:

1) During the training process, unsupervised anomaly detection requires minimal labeled attack or malicious data, which are typically more sparse and difficult to obtain.
2) Unsupervised anomaly detection does not rely on any known threat patterns, thereby performing exceptionally well when facing unknown attacks and novel threats.

However, despite the excellent performance of these models, many machine learning approaches, particularly deep learning-based unsupervised models, often exhibit a "black-box" nature, leading to a lack of interpretability and raising widespread trust issues. Many local explanation methods [11]–[15] attempt to interpret the models by presenting feature importance for individual decision points. While these local explanations can provide valuable insights for specific instances, they may not fully meet the needs of certain security applications. These fields typically require a more comprehensive understanding of how the model behaves across the entire decision space to detect complex or unknown attacks in a timely manner. In contrast, global explanations, particularly those utilizing rule extraction to describe the overall decision boundaries, may be more ideal in security systems. This is especially important as they offer the following benefits (illustrated in Figure 1):



(a) Enhancing trust in high-risk scenarios.

(b) Improving online defense capabilities.

Fig. 1: Motivation of interpreting black-box anomaly detection models into rules in security domains.

Ruoyu Li and Laizhong Cui are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518061, China (e-mail: liry@szu.edu.cn, cuilz@szu.edu.cn).

Yu Zhang is with Tsinghua University, Beijing, China, and also with Shanghai Artificial Intelligence Laboratory, Shanghai, China (e-mail: yu-zhang23@mails.tsinghua.edu.cn).

Qing Li is with the Department of Strategic and Advanced Interdisciplinary Research, Peng Cheng Laboratory, Shenzhen, China (e-mail: liq@pcl.ac.cn).

Nengwu Wu is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China (e-mail: wnw2597@gmail.com).

Yong Jiang is with the Shenzhen International Graduate School, Tsinghua University, Shenzhen, China, and also with the Department of Strategic and Advanced Interdisciplinary Research, Peng Cheng Laboratory, Shenzhen, China (e-mail: jiangy@sz.tsinghua.edu.cn).

Weizhi Meng is with the School of Computing and Communications, Lancaster University, Lancaster, UK (e-mail: weizhi.meng@ieee.org.

* The first two authors have equal contributions.

Corresponding author: Laizhong Cui.

Enhancing Trust in High-Risk Scenarios: In high-risk fields such as network intrusion detection, security operators tend to rely more on interpretable rules rather than the intuitive outputs of black-box models. Compared to complex prediction scores, interpretable rules are easier to understand and trust, which can effectively reduce false positives and mitigate potential security risks.

Improving Online Defense Capabilities: By converting the outputs of black-box models into high-fidelity rules, it becomes possible to seamlessly integrate with most rule-driven defense tools (e.g., iptables [16] and Snort [17]), ensuring efficient deployment of online defenses and efficiency in handling high-volume network traffic.

Most existing global ML/DL explanation approaches have been developed for supervised models, with limited research focused on explaining unsupervised anomaly detection. This area presents several unique challenges:

Unlabeled One-class Data (CH1). Supervised explanation methods [18]–[20] require labeled data from both positive and negative classes to determine the decision boundaries of black-box models. However, this requirement conflicts with the nature of unsupervised anomaly detection, which does not rely on labeled attack data—one of its primary advantages. This lack of labeled abnormal data adds complexity to determining the model's decision boundaries.

Lack of Surrogate Models (CH2). Global explanation methods typically use interpretable models, such as decision trees [19], [20] or linear models [21], to approximate the behavior of black-box models. However, there is a lack of suitable surrogate models in the unsupervised domain that can provide both interpretability and high performance, particularly in high-stakes security applications.

Accuracy Loss (CH3). A common issue with global methods is the loss of accuracy in the surrogate model due to simplifications made in replicating the original model [22]. While these methods can offer interpretability, they may fail to meet the high detection accuracy requirements necessary for real-time deployment in security applications.

We observe that simple surrogate models often struggle to effectively replicate the complexity of black-box models when dealing with high-dimensional data. Such instances belonging to the same class may contain multiple underlying distributions, with the overall distribution composed of several sub-distributions. For example, the normal activities of a host may involve multiple services (such as web services, database services), each exhibiting distinctly different characteristics in feature space. This diversity makes it challenging for simple surrogate models to fully capture the complex behaviors of black-box models, thereby reducing the interpretability and accuracy of the model.

Conclusion: To this end, this paper proposes a divide-and-conquer method called Ano2Rule to extract global rules from black-box unsupervised anomaly detection models, aiming to provide interpretable, accurate, and robust explanations. The implementation of Ano2Rule relies on two core components: the **Interior Clustering Tree (IC-Tree)** and the **Compositional Boundary Exploration (CBE)** algorithm. The IC-Tree is an unsupervised decision tree model that extends the traditional Classification And Regression Tree (CART) by incorporating the predictions of black-box models to determine data splits. It partitions the high-dimensional data space into smaller, more manageable subspaces, capturing complex distributions and identifying normal behavior patterns, thereby laying the foundation for explaining the black-box model. Within each IC-Tree subspace, the CBE algorithm generates boundary inference rules. Starting from initial hypercube-shaped rules, CBE gradually optimizes the decision boundaries through an approximate gradient ascent method, accurately capturing the black-box model's complex boundaries. By combining the rule sets generated by the IC-Tree and CBE, Ano2Rule fully presents the global decision logic of the black-box model, ensuring high fidelity of the explanatory rules.

Beyond providing global explanations, Ano2Rule also exhibits strong extensibility to other types of model interpretation, such as **local explanation** and **counterfactual explanation**. This flexibility is rooted in the explicit, human-understandable characterization of normal boundaries from the perspective of the black-box model. Specifically, for local explanation (i.e., assessing feature importance for individual predictions), the significance of each feature can be quantified by measuring the distance between a sample's feature values and the boundaries defined by its matched rule. For counterfactual explanation (i.e., determining how an anomalous sample could be modified to be classified as normal), our method can project the anomaly onto the closest region defined by the rules and optimize the necessary feature changes accordingly, thereby generating informative counterfactuals. These extensions enable our approach to support a versatile suite of explanation tasks with unified technical foundations.

For evaluation, we used four unsupervised anomaly detection models and trained them on four benchmark datasets. To comprehensively assess the effectiveness of our method, we compared it with five baseline explanation methods. The experimental results show that Ano2Rule not only can extract highly interpretable and precise rules from black-box models but also outperforms existing methods in multiple metrics, including fidelity, robustness, true positive rate (TPR), and true negative rate (TNR), meeting the demand of improving human trust in black-box models and maintaining high detection accuracy for online deployment. We also validate the extensibility of Ano2Rule by demonstrating its capability to provide local and counterfactual explanations. Specifically, we show that our rule-based approach enables quantifiable local feature importance and generates actionable counterfactuals for anomalous samples, further illustrating the versatility and practical value of the proposed method.

The remainder of this paper is structured as follows: Section II reviews related work on unsupervised anomaly detection and model interpretability. Section III outlines the objectives of the proposed method, introducing distribution decomposition rules and boundary inference. Sections IV detail the core algorithms of Ano2Rule. Section V presents the theoretical analysis of the proposed method. Section VI elaborates on the extensibility of AnoRule to other types of explanation. Section VII describes the experimental setup and result analysis. Section VIII concludes with key findings and suggests future research directions.

II. Related Work

A. Anomaly Detection in Security

Anomaly detection is a cornerstone technique in the field of cybersecurity, supporting a wide range of applications including network intrusion detection [1]–[3], malware detection [4], [5], [23], and IoT security [8], [9], [24], [25]. The primary objective is to identify rare or previously unseen events—often corresponding to malicious activities—that deviate from the established patterns of normal system behavior. This is especially important in modern security environments, where the diversity and sophistication of threats continue to evolve rapidly, making it impractical to enumerate all possible attack types in advance.

Traditional signature-based detection systems rely on predefined patterns and can only recognize known threats. In contrast, anomaly detection offers the ability to discover novel or zero-day attacks by flagging deviations as potential anomalies. In practical settings, the abundance of benign data and the scarcity of labeled attack samples further motivate the adoption of unsupervised anomaly detection models, which can learn normal behavior from unlabeled data without requiring explicit attack labels. Recent years have witnessed significant progress in unsupervised anomaly detection methods for security applications. Representative approaches include one-class classifiers such as One-Class SVM (OCSVM) [26]–[28], tree-based models like Isolation Forest (iForest) [29], [30], and neural network-based techniques such as autoencoders (AE) and variational autoencoders (VAE) [31].

Despite these advances, the widespread adoption of anomaly detection in high-stakes security contexts remains hindered by the lack of interpretability and transparency in modern machine learning models. Security operators must be able to understand and trust the decisions made by automated systems, especially when incorrect predictions could result in substantial operational or financial losses [15]. As a result, enhancing the interpretability of anomaly detection models has become a research priority in both academia and industry.

B. Model Interpretation

To address this issue, explainable artificial intelligence (XAI) has gradually been introduced into the anomaly detection field to bridge the semantic gap between model predictions and human understanding. Existing XAI methods can be categorized into model-specific explanation methods and model-agnostic explanation methods.

1) Model-Specific Explanation: To address this issue, explainable artificial intelligence (XAI) has gradually been introduced into the anomaly detection field to bridge the semantic gap between model predictions and human understanding [32]–[35]. Existing XAI methods can be categorized into model-specific explanation methods and model-agnostic explanation methods. Some studies attempt to integrate explainability directly into model architectures to enhance interpretability. For example, Kauffmann et al. proposed a decomposition method for explaining one-class support vector machine (SVM) anomaly detection [32], breaking down the model's decision-making process into comprehensible components. Aguilar et al. proposed an interpretable autoencoder based on decision trees, which provides a natural explanation for experts in the application area [35]. Feng et al. utilized a k-dimensional (KD) tree to classify DDoS sources at an IP-level fine granularity, outputting explanatory information that enables easy inspection of detection results [36]. However, such approaches are often tightly coupled with the underlying model and lack generality, limiting their application to a broader range of black-box anomaly detectors.

2) Local Explanation: Unlike model-specific methods that require internal access to model structures or parameters, model-agnostic techniques treat the target model as a black box and seek to explain its predictions solely based on input-output behavior. Local explanation, as a representative approach, aims to interpret a model's prediction for a specific input by quantifying the importance of each feature in the decision-making process. Among the most prominent local explanation frameworks are LIME (Local Interpretable Model-agnostic Explanations) [11] and SHAP (SHapley Additive exPlanations) [12]. These methods operate by constructing simple, interpretable surrogate models—such as sparse linear regressors or decision trees—in the vicinity of the input instance under consideration. For example, LIME perturbs the features of a sample to generate synthetic neighbors and fits a local model to approximate the original black-box model's behavior around that point. These methods have been applied to explain various unsupervised models [37]–[39].

In the security domain, local explanations have proved valuable for investigating individual alerts or suspicious activities detected by anomaly detection systems. For instance, Guo et al. leveraged local interpretable models to analyze malware classification results, offering actionable insights for analysts to prioritize incident response [14]. Similarly, Sipple applied integrated gradients [40], a local explanation technique, to trace root causes in IoT device failures [41].

However, local explanations face several limitations in practice. They may be sensitive to the definition of the local neighborhood, potentially resulting in unstable or misleading feature attributions. Furthermore, while local methods can reveal "why" for individual cases, they often fail to capture the broader decision logic or recurring patterns that are essential for proactive defense and policy-making.

3) Counterfactual Explanation: The central idea of counterfactual explanation is to answer "what if" questions by identifying the minimal changes needed to an input instance in order to alter the model's prediction. For instance, in anomaly detection, a counterfactual explanation provides insight into how an anomalous instance could be transformed into a normal one, offering clear and actionable guidance to analysts.

The general methodology for generating counterfactuals involves solving an optimization problem to find the closest data point whose predicted class is different from the original. This paradigm was formalized by Wachter et al., who proposed counterfactual explanations as a means to interpret automated decisions while treating the underlying model as a black box [42]. Extensions to this idea have focused on generating diverse counterfactuals for greater coverage and robustness, as demonstrated by DiCE [43]. Counterfactual methods are also developed for complex data structures, such as graphs [44] and

TABLE I: Comparison of explanation methods for anomaly detection; * local methods may provide limited counterfactual samples via sampling or perturbation, but are not designed for minimal changes; (✓) indicates partial or limited support.

Method	Comprehensive Decision Logic	Instance-level Attribution	Actionable Guidance	Unsupervised Adaptation	High Fidelity/ Robustness
Model-Specific (e.g., [32], [34])	✓	×	×	×	✓
Local (e.g., [11], [12])	×	✓	(✓)*	×	×
Counterfactual (e.g., [42], [43])	×	×	✓	×	×
Surrogate Trees/Rules (e.g., [20], [48])	✓	×	×	(✓)	(✓)
Ano2Rule (Ours)	✓	✓	✓	✓	✓

recommender systems [45].

In the field of security and anomaly detection, counterfactual explanations are emerging as a valuable tool for operational transparency and actionable insight. For example, Guzman et al. explored counterfactual explanations for detecting SQL injection attacks, helping analysts understand how an attack instance might be altered to avoid detection or be classified as benign [46]. Similarly, counterfactuals can provide interpretable recommendations for revising loan applications in a security-critical financial context [47].

Despite this progress, applying counterfactual explanations to unsupervised anomaly detection remains challenging, due to the absence of clear class boundaries and the need to preserve data realism. Existing security-related counterfactual approaches typically focus on supervised settings and require labeled data.

4) Global Explanation: Global methods are designed to provide a comprehensive, human-understandable summary of a model's overall decision-making process across the entire input space. Global interpretability is particularly important for security operators, who must not only investigate individual alerts but also understand and audit system-wide detection logic to ensure reliable deployment, compliance, and risk mitigation.

A common strategy for global explanation is to approximate the black-box model with an inherently interpretable surrogate, such as decision trees [19], [20], [49], symbolic rules [18], sparse linear models [21], and decision lists [50], to approximate and explain the predictions of any well-trained model. In related research, authors have constructed global explanations in the form of decision trees to provide insight into black-box models [19]. The framework proposed by Jacobs et al. generates tree models using existing machine learning models and training datasets to explain security-related decisions [20].

However, most of these global explanation approaches have been developed in supervised learning settings, where labeled examples of both normal and anomalous data are available. This limits their direct applicability to the unsupervised anomaly detection scenarios, where only normal data is typically used for training. Additionally, while some works can extract rules from unsupervised anomaly detection models [48], they still assume that the training dataset contains sufficient outliers to determine decision boundaries, which may not hold true in practice due to the model's generalization capabilities.

Recent studies have also attempted to aggregate multiple local explanation models to provide nearly global explanations [51]–[53]. However, these approaches incur high computational costs when processing large-scale data and require a trade-off between fidelity and coverage. Although techniques such as knowledge distillation can reduce complexity and enhance explainability through model transformation [54], [55], their primary goal remains model compression and accuracy assurance, rather than providing high-fidelity explanations of the original models. Consequently, achieving comprehensive interpretability for unsupervised anomaly detection while maintaining high detection performance remains a challenge.

In summary, while existing methods each address important aspects, they often suffer from limited generality, restricted support for unsupervised settings, or lack of high-fidelity explanations. As summarized in Table I, our proposed rule-based framework stands out by simultaneously offering comprehensive decision logic, instance-level attribution, and actionable guidance, all within a robust and unsupervised-compatible paradigm, which bridges gaps in interpretability for security-focused anomaly detection.

III. Overview

To tackle the interpretability challenges in unsupervised anomaly detection within security contexts, we introduce a novel framework called Ano2Rule, which is designed to extract interpretable, rule-based explanations that align closely with the original anomaly detection model, enhancing transparency and operational efficacy. As shown in Figure 2, it leverages an Interior Clustering Tree (IC-Tree) for feature space partitioning and a Compositional Boundary Exploration (CBE) algorithm for precise decision boundary inference.

A. Problem Definition

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the variable space of d-dimensional features; $\boldsymbol{x}$ and x_i denote a data sample and its i-th dimension of features. We give the following definitions for the rest of the paper:

Definition 1 (Unsupervised Anomaly Detection). *Given unlabeled negative (normal) data $\boldsymbol{X}$ sampled from a stationary distribution $\mathcal{D}$ for training, an unsupervised model estimates the probability density function $f(\boldsymbol{x}) \approx \mathrm{P}_{\mathcal{X}\sim\mathcal{D}}(\boldsymbol{x})$, and detects an anomaly via a low probability $f(\boldsymbol{x}) < \varphi$, where $\varphi > 0$ is a threshold determined by the model itself or by humans.*

It is noted that the threshold φ is a non-zero value, meaning that the model inevitably generates false positives, which is a common setting in most of the works [1]–[3] even though the false positive rate can be very low. Besides, the normal data may occasionally be contaminated or handled with errors. We consider the anomaly detection tolerant of noisy data, but their proportion in the training dataset is small and we do not have deterministic labels of the training data.

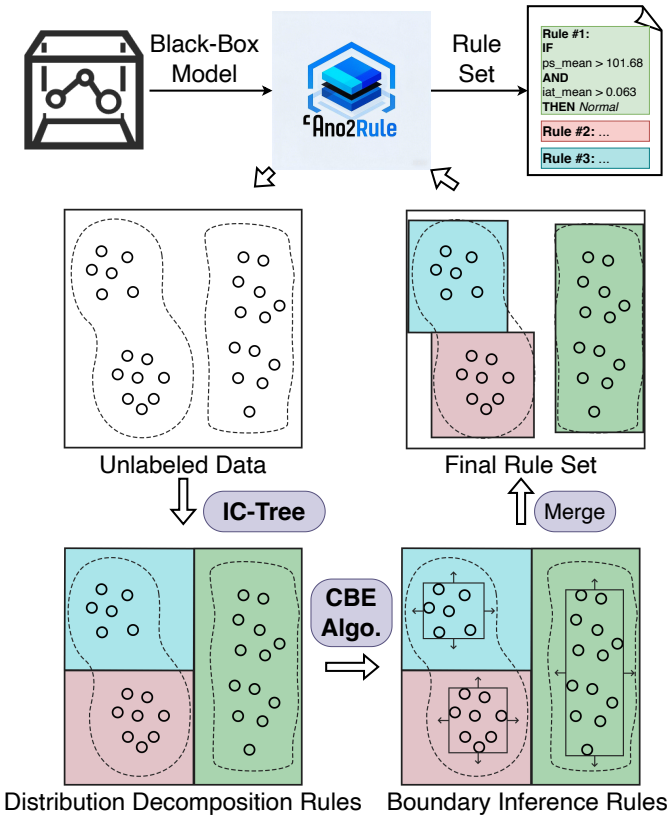


Fig. 2: A high-level illustration of Ano2Rule; small circles are unlabeled normal data; dashed curves are the decision boundary of the black-box model; vertical/horizontal lines are extracted rules at each step.

Definition 2 (Global Explanation by Rule Extraction). *Given a trained model f with its anomaly threshold φ and the training set $\boldsymbol{X}$, we obtain an in-distribution rule set $\mathcal{C} = \{C_1, C_2, ...\}$ that explains how the model f profiles the distribution of normal data. A rule $C = ...\wedge(x_i \odot \upsilon_i)\wedge...\wedge(x_j \odot \upsilon_j)$ is a conjunction of several axis-aligned constraints on a subset of the feature space, where υ_i is the bound for the i-th dimension and $\odot \in \{\leq, >\}$.*

Let $\boldsymbol{x} \in C$ indicate that a data sample satisfies a rule. From $\mathcal{C}$, we can build a surrogate model $h_{\mathcal{C}}(\boldsymbol{x})$, whose inference is to regard a data sample that cannot match any of the extracted rules as anomalous:

$$h_{\mathcal{C}}(\boldsymbol{x}) = \neg(\boldsymbol{x} \in C_1) \wedge \neg(\boldsymbol{x} \in C_2) \wedge ..., \; C_i \in \mathcal{C}. \quad (1)$$

Design Goal. We expect the extracted rules to have a high fidelity to the original model, that is, a similar coverage of normal data (i.e., true negative rate), and a similar detection rate of anomalies (i.e., true positive rate). We formulate our objective as follows:

$$\arg\min_{\mathcal{C}} \mathcal{L}_{\mathcal{X}\sim\mathcal{D}}(\mathcal{C}, f, \varphi) + \mathcal{L}_{\mathcal{X}\nsim\mathcal{D}}(\mathcal{C}, f, \varphi). \quad (2)$$

B. Methodology Overview

To minimize the first item in Equation (2), suppose the training data $\boldsymbol{X}$ can well represent the distribution $\mathcal{D}$, a straightforward approach is to find the bound of $\boldsymbol{X}$ as rules, such as using a hypercube to enclose the data samples which can easily achieve the minimization of the partial loss $\mathcal{L}_{\boldsymbol{x}\in\boldsymbol{X}}(\mathcal{C}, f, \varphi) = 0$. However, as $\mathcal{D}$ is not a prior distribution and we do not have labeled abnormal samples, the second item $\mathcal{L}_{\mathcal{X}\nsim\mathcal{D}}(\mathcal{C}, f, \varphi)$ is neither deterministic nor estimable unless we create sufficient random samples and query f, which is challenging given the high-dimensional space of $\mathcal{X}$.

As prior studies [25], [41] suggest, normal data are typically multimodal, i.e., the overall distribution is formed by multiple compositional distributions. For example, a server supports multiple services such as web, email and database. The representations of these services can be disparate and located in different regions in feature space with little transition between the regions, making it infeasible to find a uniform rule set to accurately estimate the original model.

Based on this intuition, Ano2Rule adopts a divide-and-conquer approach, as shown in Figure 2. First, we propose an *Interior Clustering Tree* (IC-Tree) to find the *distribution decomposition rules*, which cut the feature space into subspaces so that each subspace contains data belonging to the same compositional distribution. Then, we design a *Compositional Boundary Exploration* algorithm (CBE) to explore the decision boundary on each compositional distribution. Particularly, the algorithm starts from the minimal hypercube that encloses all data of the distribution, and finds the boundary by recursively extending the boundary following the optimal direction guided by a gradient approximation. Upon obtaining the decision boundary of a distribution, the corresponding *boundary inference rule* can be extracted. Last, the final rule set can be obtained by merging the distribution decomposition rule and the boundary inference rule of each compositional distribution. We formally define the distribution decomposition rule and the boundary inference rule as follows:

Definition 3 (Distribution Decomposition Rule). *Denoted by C_k^I that decomposes the overall distribution of normal data $\mathcal{D}$ into K compositional distributions, i.e., $\mathrm{P}_{\mathcal{X}\sim\mathcal{D}}(\boldsymbol{x}) = \sum_{k=1}^{K} \phi_k \cdot \mathrm{P}_{\mathcal{X}\sim\mathcal{D}_k}(\boldsymbol{x}|\boldsymbol{x} \in C_k^I)$ where ϕ_k denotes the weight of each compositional distribution, so that a data sample $\boldsymbol{x} \sim \mathcal{D}_k$ has significantly small probability of belonging to other distributions.*

Definition 4 (Boundary Inference Rule). *Denoted by C_k^E that estimates the decision boundary of the original model for each distribution $\mathcal{D}_k$, i.e., $\arg\min_{C_k^E} \mathcal{L}_{\mathcal{X}\sim\mathcal{D}_k}(C_k^E, f, \varphi) + \mathcal{L}_{\mathcal{X}\nsim\mathcal{D}_k}(C_k^E, f, \varphi)$.*

With the definition of these two types of rules, we translate the objective in Equation (2) to the following objective as our intuition indicates.

Proposition 1. *The original objective can be estimated by finding the union of the conjunction of distribution decomposition rules and boundary inference rules for each compositional distribution:*

$$\bigcup_{k=1}^{K} \arg\min_{C_k} \mathcal{L}_{\mathcal{X}\sim\mathcal{D}_k}(C_k, f, \varphi) + \mathcal{L}_{\mathcal{X}\nsim\mathcal{D}_k}(C_k, f, \varphi), \quad (3)$$

where $C_k = C_k^I \wedge C_k^E$.

IV. Design Details

A. Interior Clustering Tree

To derive the distribution decomposition rules and partition the feature space into distinct subspaces, it is essential to first define the compositional distribution D_k. In unsupervised anomaly detection, the lack of labeled data makes it challenging to directly infer decision boundaries. By defining D_k, we can decompose the complex overall distribution D into interpretable sub-distributions, enabling a rule-based description of the model's decision-making process. Although labeled data is unavailable, the estimation of the overall distribution D from the outputs of the black-box model can serve as a basis for this decomposition.

Suppose two data samples $x^{(i)}$ and $x^{(j)}$ belong to the same distribution D_k. In that case, the difference in their probabilities of belonging to the overall distribution D will be less than a small constant ϵ. Based on the definition of the distribution decomposition rules and compositional distributions, if these two data samples belong to the same distribution D_k, their probability of belonging to any other distribution $P_{X\sim D_l}(x)$ (where $l \neq k$) is nearly zero. Consequently, the probability of these samples belonging to the overall distribution $P_{X\sim D}(x)$, which is a weighted sum of the probabilities across all compositional distributions, is approximately equal to their probability of belonging to $P_{X\sim D_k}(x)$.

Based on this, we propose a tree-based model dubbed Interior Clustering Tree (IC-Tree), which extends the CART decision tree [56]. The main difference between IC-Tree and CART is that, rather than splitting data based on ground truth labels, IC-Tree uses the probability output by the original model as splitting criteria, enabling it to work in a completely unsupervised manner. This decomposition approach provides a clearer understanding of how different regions of the feature space correspond to the model's decision-making logic, enhancing both transparency and interpretability.

Node Splitting. Given the data $\boldsymbol{N}$ at a tree node, we first obtain the output of the anomaly detection model $f(\boldsymbol{x})$ for $\boldsymbol{x} \in \boldsymbol{N}$. Similar to decision trees, the node of an IC-Tree finds a splitting point $s = (i, b_i)$ that maximizes the gain:

$$s = \arg\max_{s} I(\boldsymbol{N}) - \frac{|\boldsymbol{N}_l|}{|\boldsymbol{N}|} I(\boldsymbol{N}_l) - \frac{|\boldsymbol{N}_r|}{|\boldsymbol{N}|} I(\boldsymbol{N}_r), \quad (4)$$

where b_i is the splitting value for the i-th dimension, $\boldsymbol{N}_l$ and $\boldsymbol{N}_r$ are the data split to the left and right child nodes, $|\boldsymbol{N}|$ denotes the number of data samples, and I is a criterion function such as Gini index $I = 2p(1-p)$ for binary classification with the probability of p. Specifically, we let p be the average output of the anomaly detection model, which can be interpreted as the expectation of the probability that the data belong to the same distribution:

$$p = \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{N}}[\mathrm{P}_{\mathcal{X}\sim\mathcal{D}}(\boldsymbol{x})] = \frac{1}{|\boldsymbol{N}|} \sum_{\boldsymbol{x}\in\boldsymbol{N}} f(\boldsymbol{x}). \quad (5)$$

An IC-Tree continues to split nodes until it satisfies one of the following conditions: i) the number of data samples at the node $|\boldsymbol{N}| = 1$; ii) for any two of the data samples at the node $\forall \boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \in \boldsymbol{N}$, $|f(\boldsymbol{x}^{(i)}) - f(\boldsymbol{x}^{(j)})| < \epsilon$; iii) it reaches a maximum depth τ, which is a hyperparameter. The IC-Tree focuses on the subset of features that contribute the most to the variability of anomaly scores at each split, thereby ensuring a more accurate separation of data distribution.

Distribution Decomposition Rule Extraction. A trained IC-Tree that has K leaf nodes ($K \leq 2^\tau$) represents K distributions separated from the overall distribution $\mathcal{D}$. Suppose the k-th leaf node has a depth of τ'. A distribution decomposition rule that describes the k-th compositional distribution can be extracted by the conjunction of the splitting constraints from the root to the leaf node:

$$C_k^I = (x_i \odot_1 b_i | s_1 = (i, b_i)) \wedge ... \wedge (x_j \odot_{\tau'} b_j | s_{\tau'} = (j, b_j)), \quad (6)$$

where $\odot$ is "$\leq$" if the decision path goes left or "$>$" if the decision path goes right.Each rule set C_k^I describes the conditions of the k-th sub-distribution in the feature space. These rules enable the accurate classification of new data points into their corresponding sub-distributions.

B. Compositional Boundary Exploration

Following the generation of distinct subspaces by the IC-Tree, we propose the Compositional Boundary Exploration (CBE) algorithm (described in Algorithm 1) to precisely define and refine the decision boundaries within each subspace. The CBE algorithm uses the minimal hypercube that encloses the normal data of each compositional distribution as a starting point. Further, we refer to adversarial attacks [57] and propose a method to approximate the optimal direction to explore the decision boundary, which makes the algorithm more efficient and accurate to estimate the decision boundary.

Starting from Hypercube (line 1). Let $\boldsymbol{X}_k$ denote the training data falling into the k-th leaf node of an IC-Tree that represents a compositional distribution. Recall the definition of boundary inference rules that target $\min \mathcal{L}_{\mathcal{X}\sim\mathcal{D}_k}(C_k^E, f, \varphi) + \mathcal{L}_{\mathcal{X}\nsim\mathcal{D}_k}(C_k^E, f, \varphi)$. We use the minimal hypercube H_k as a starting point of boundary inference rules to bound every dimension of the data samples in $\boldsymbol{X}_k$ judged by the original model as normal, which obviously achieves $\mathcal{L}_{\boldsymbol{x}\in\boldsymbol{X}_k}(H_k, f, \varphi) = 0$. The minimal hypercube is enclosed by $2 \times d$ axis-aligned hyperplanes, which can be characterized by the following rule:

$$H_k = (v_1^- \leq x_1 \leq v_1^+) \wedge ... \wedge (v_d^- \leq x_d \leq v_d^+), \quad (7)$$

where $v_i^- = \min(x_i | f(\boldsymbol{x}) > \varphi, \boldsymbol{x} \in \boldsymbol{X}_k)$ and $v_i^+ = \max(x_i | f(\boldsymbol{x}) > \varphi, \boldsymbol{x} \in \boldsymbol{X}_k)$.

Explorer Sampling (line 4∼6. The CBE algorithm explores the decision boundary of the original model by estimating the bound of one feature dimension at a time. For i-th dimension, we uniformly sample N_e data points on each hyperplane of the hypercube, i.e., $\boldsymbol{e}^{(1)}, ..., \boldsymbol{e}^{(N_e)} \in H_k \wedge (x_i = v_i), v_i \in \{v_i^-, v_i^+\}$, which are called the *initial explorers* for this hyperplane. For an initial explorer $\boldsymbol{e}$, we further sample N_s *auxiliary explorers* near it from a truncated multivariant Gaussian distribution denoted by $\mathcal{N}(\boldsymbol{e}, \boldsymbol{\Sigma}, i)$. Particularly, the center of sampling is the explorer $\boldsymbol{e}$ and the radius of sampling is constrained by the sampling matrix $\boldsymbol{\Sigma} = \mathrm{diag}(\rho|v_1^+ - v_1^-|, ..., \rho|v_d^+ - v_d^-|)$, where ρ is a hyperparameter, and the sampling on i-th dimension is half-truncated to only keep the distribution outside the

hypercube as we desire to extend the boundary. With $N_e \times N_s$ auxiliary explorers in total, we query the original model and use Beam Search to select N_e samples with the minimal probability of being normal as the candidate explorers for the next iteration.

Gradient Approximation (line 7∼9). Though we have obtained N_e candidate explorers in the previous step, using them directly for the next iteration does not guarantee the optimal direction of movement towards the decision boundary. To find the optimal direction, we utilize the Fast Gradient Sign Method [57] that employs gradient ascent to find the direction of feature perturbation. However, we do not know the loss function of the original model in black-box scenarios. To deal with it, given a selected auxiliary explorer $\hat{\boldsymbol{e}}$ that is sampled around an initial explorer $\boldsymbol{e}$ on the i-th dimension hyperplane, we approximate the i-th dimension of the model gradient (i.e., the partial derivative) by the slope of a linear model across the two data points, and use the midpoint with its i-th dimension minus the approximation as the new explorer for the next iteration:

$$e_{i,\ \text{next}} = \frac{e_i + \hat{e}_i}{2} - \eta \cdot \text{sign}(\nabla_i), \tag{8}$$

$$\nabla_i = \frac{\partial f(\boldsymbol{x})}{\partial x_i} \approx \frac{f(\boldsymbol{e}) - f(\hat{\boldsymbol{e}})}{e_i - \hat{e}_i}, \tag{9}$$

where $sign(\cdot)$ is the sign function, and η is a hyperparameter to control the stride of one iteration. The iteration stops when i) an auxiliary explorer $\hat{\boldsymbol{e}}_{ext}$ that satisfies $f(\hat{\boldsymbol{e}}_{ext}) < \varphi$ is found, or ii) it reaches the maximum number of iterations.

Rule Acquisition (line 12). Once the iteration stops, boundary constraints are generated for each dimension.If the iteration stops due to the first condition, we produce a boundary constraint for each dimension using the coordinate of $\hat{\boldsymbol{e}}_{ext}$ that extends the boundary of the hypercube, i.e., $c_i = (x_i \odot \hat{e}_{ext,i})$, where $\odot$ is "$\leq$" if $\hat{e}_{ext,i}$ is greater than υ_i^+, or "$>$" if $\hat{e}_{ext,i}$ is less than υ_i^-. If the iteration stops due to the second condition, it means the algorithm encounters difficulties in moving towards the decision boundary by perturbing this feature dimension. We calculate the difference between the model prediction of the last auxiliary explorer and that of the initial explorers on the hyperplane. If the difference is smaller than a threshold δ, we decide that this feature dimension is a *contour line*, i.e., it has no significant correlation with the model prediction. In this case, we do not produce any constraints for this dimension. If the difference is greater than the threshold, we produce constraints in the same way as those produced under the first condition. The final boundary inference rule is the disjunction of the hypercube and the constraints on each dimension.

C. Merging Final Rule Set

When the stopping condition is met, we merge the distribution decomposition rules and boundary inference rules on each compositional distribution by their conjunction, and the final rule set is the union of the extracted rules:

$$\mathcal{C} = \bigcup_{k=1}^{K} C_k, \text{ where } C_k = C_k^I \wedge C_k^E. \tag{10}$$

We present the complete process of Ano2Rule in Algorithm 2. Generally, the extracted rules serve as an "allowlist", describing the range of multiple features that can be considered normal and judging others as anomalies, such as:

If $x_{j_1} < \theta_1$ **and** $x_{j_2} \geq \theta_2$, **then normal**

Algorithm 1: Compositional Boundary Exploration

Input: Data falling into the k-th leaf node $\boldsymbol{X}_k$, anomaly detector f and its threshold φ
Output: Boundary inference rule C_k on this leaf node such that C_k encapsulates normality

1 $H_k \leftarrow$ MinimalHypercube(X_k);
2 **for** i-th dimension **in** $\boldsymbol{X}_k$ **do**
3 $\quad \boldsymbol{e}^{(1)}, ..., \boldsymbol{e}^{(N_e)} \leftarrow$ IntialExplorer(H_k) on i-th dimension;
4 $\quad$ **while** True **do**
5 $\quad\quad \hat{\boldsymbol{e}}^{(1)}, ..., \hat{\boldsymbol{e}}^{(N_s)} \leftarrow$ AuxiliaryExplorer$(\boldsymbol{e})$ for each initial explorer $\boldsymbol{e}$;
6 $\quad\quad$ Beam Search for N_e candidate explorers from $N_e \times N_s$ auxiliary explorers that have the minimal probability of being normal judged by f and φ;
7 $\quad\quad \boldsymbol{e} \leftarrow$ GradientApprox$(\hat{\boldsymbol{e}})$ for each candidate explorer selected from auxiliary explorers;
8 $\quad\quad$ **if** ending condition satisfied **then**
9 $\quad\quad\quad c_i \leftarrow (x_i \odot \hat{e}_i)$ and **break**;
10 $\quad$ **end while**
11 **end for**
12 **return** $C_k^E = H_k \vee (c_1 \wedge c_2 \wedge ... \wedge c_d)$;

Algorithm 2: Rule Extraction Process of Ano2Rule

Input: Unlabeled data $\boldsymbol{X}$, anomaly detector f and its threshold ϕ
Output: In-distribution rule set $\mathcal{C}$ as interpretation for f

1 $\{\boldsymbol{N}_k\}_{k=1}^K \leftarrow$ IC-Tree$(\boldsymbol{X}, f; \tau)$;
2 **for** $\boldsymbol{N}_k$ **in** $\{\boldsymbol{N}_k\}_{k=1}^K$ **do**
3 $\quad C_k^I \leftarrow$ getRule$(\boldsymbol{N}_k)$; `// get its distribution decomposition rule`
4 $\quad \boldsymbol{X}_k \leftarrow$ getData$(\boldsymbol{N}_k)$;
5 $\quad C_k^E \leftarrow$ CBE$(\boldsymbol{X}_k, f, \phi)$; `// get its boundary inference rule`
6 $\quad C_k \leftarrow C_k^I \wedge C_k^E$;
7 **end for**
8 **return** $\mathcal{C} = \bigcup_{k=1}^K C_k$;

V. Theoretical Analysis

A. Proof of Proposition 1

One of the key claims that lay the theoretical foundation for our method is Proposition 1, which employs a divide-and-conquer approach by translating the objective in Equation (2) to Equation (3). We will prove that such an approach is feasible: the extracted rules will have a high fidelity to the original model, that is, a similar coverage of normal data (i.e., true

negative rate, TNR), and a similar detection rate of anomalies (i.e., true positive rate, TPR).

We prove the following lemma equivalent to Proposition 1:

Lemma 1. *If the distribution decomposition rules and the inference boundary rules that minimize the loss on each of the compositional distributions are found, the sum of the minimum losses on each of the compositional distributions can estimate the minimum loss on the overall distribution with a significantly small error ψ, i.e.,*

$$\begin{aligned}&\min \mathcal{L}_{\mathcal{X}\sim\mathcal{D}}(\mathcal{C},f,\varphi)+\mathcal{L}_{\mathcal{X}\nsim\mathcal{D}}(\mathcal{C},f,\varphi)\\ =&\sum_{k=1}^{K}\min(\mathcal{L}_{\mathcal{X}\sim\mathcal{D}_k}(C_k,f,\varphi)+\mathcal{L}_{\mathcal{X}\nsim\mathcal{D}_k}(C_k,f,\varphi))+\psi,\\ &\text{where } C_k=C_k^I\wedge C_k^E,\psi\geq 0.\end{aligned} \tag{11}$$

Proof. The sum of the minimum losses on each of the compositional distributions is calculated by an iteratively cumulative process. Let L_j be the sum of the minimum losses on each of the compositional distributions at the j-th iteration:

$$L_j=\sum_{k=1}^{j}\min(\mathcal{L}_{\mathcal{X}\sim\mathcal{D}_k}(C_k,f,\varphi)+\mathcal{L}_{\mathcal{X}\nsim\mathcal{D}_k}(C_k,f,\varphi)). \tag{12}$$

Let $\mathcal{X}\sim\bigcup_{k=1}^{j}\mathcal{D}_k$ represent a variable belonging to any of the compositional distributions $\mathcal{D}_1,...,\mathcal{D}_j$, which is the same as the overall distribution $\mathcal{X}\sim\mathcal{D}$. We prove the Loop Invariant of L_j during the iteration, which always satisfies:

$$\begin{aligned}&L_j=\min(\mathcal{L}_{TNR}^j+\mathcal{L}_{TPR}^j)-\psi,\\ &\text{where } \mathcal{L}_{TNR}^j=\mathcal{L}_{\mathcal{X}\sim\bigcup_{k=1}^{j}\mathcal{D}_k}(\bigcup_{k=1}^{j}C_k,f,\varphi),\\ &\mathcal{L}_{TPR}^j=\mathcal{L}_{\mathcal{X}\nsim\bigcup_{k=1}^{j}\mathcal{D}_k}(\bigcup_{k=1}^{j}C_k,f,\varphi)\end{aligned} \tag{13}$$

1) For the first iteration,

$$\begin{aligned}L_1&=\min\mathcal{L}_{\mathcal{X}\sim\mathcal{D}_1}(C_1,f,\varphi)+\mathcal{L}_{\mathcal{X}\nsim\mathcal{D}_1}(C_1,f,\varphi)\\ &=\min\mathcal{L}_{\mathcal{X}\sim\mathcal{D}}(\mathcal{C},f,\varphi)+\mathcal{L}_{\mathcal{X}\nsim\mathcal{D}}(\mathcal{C},f,\varphi)-\psi\end{aligned} \tag{14}$$

obviously holds where $\psi=0$.

2) Suppose the Equation (13) holds at the j-th iteration. For the $(j+1)$-th iteration, we have the following derivations:

$$\begin{aligned}&L_{j+1}=\min(\mathcal{L}_{TNR}^j+\mathcal{L}_{TPR}^j)-\psi\\ &+\min(\mathcal{L}_{\mathcal{X}\sim\mathcal{D}_{j+1}}(C_{j+1},f,\varphi)+\mathcal{L}_{\mathcal{X}\nsim\mathcal{D}_{j+1}}(C_{j+1},f,\varphi))\\ &=\min(\mathcal{L}_{TNR}^{j+1}+\mathcal{L}_{TPR}^{j+1})-\psi\\ &+\mathcal{L}_{\mathcal{X}\sim\bigcup_{k=1}^{j}\mathcal{D}_k\cap\mathcal{D}_{j+1}}(\bigcup_{k=1}^{j+1}C_k,f,\varphi)\\ &+\mathcal{L}_{\mathcal{X}\nsim\bigcup_{k=1}^{j}\mathcal{D}_k\cap\mathcal{D}_{j+1}}(\bigcup_{k=1}^{j+1}C_k,f,\varphi),\end{aligned} \tag{15}$$

where $\bigcup_{k=1}^{j+1}\mathcal{D}_k\cap\mathcal{D}_{j+1}$ represents the overlap area between the conjunction of the compositional distributions $\bigcup_{k=1}^{j+1}\mathcal{D}_k$ and the $(j+1)$-th compositional distribution $\mathcal{D}_{j+1}$. Recall the definition of the compositional distributions that a data sample belonging to one compositional distribution has a significantly small probability of belonging to other compositional distributions, meaning that the overlap area between the compositional distributions is significantly small. Therefore, the loss with respect to the overlap area is also significantly small, given the data samples belonging to the area are significantly rare. Let

$$\begin{aligned}\psi=\psi&+\mathcal{L}_{\mathcal{X}\sim\bigcup_{k=1}^{j}\mathcal{D}_k\cap\mathcal{D}_{j+1}}(\bigcup_{k=1}^{j+1}C_k,f,\varphi)\\ &+\mathcal{L}_{\mathcal{X}\nsim\bigcup_{k=1}^{j}\mathcal{D}_k\cap\mathcal{D}_{j+1}}(\bigcup_{k=1}^{j+1}C_k,f,\varphi),\end{aligned} \tag{16}$$

and we can get the final result of L_{j+1}:

$$L_{j+1}=\min(\mathcal{L}_{TNR}^{j+1}+\mathcal{L}_{TPR}^{j+1})-\psi, \tag{17}$$

which proves the loop invariant in Equation (13). When $j=K$, as the overall distribution is equal to the conjunction of the compositional distributions, i.e., $\mathcal{D}=\bigcup_{k=1}^{K}\mathcal{D}_k$, we prove Equation (11) holds and Lemma 1 is correct.

TABLE II: Computational complexity of Ano2Rule.

Phase	Component	Complexity
Rule extraction	IC-Tree	$O(d\cdot n\log n)$
	CBE Algorithm	$O(K\cdot d\cdot N_e\cdot N_s)$
Execution	Ano2Rule	$O(\lvert C\rvert\cdot d)$

B. Computational Complexity

For rule extraction, the complexity of the IC-Tree is identical to a CART: $O(d\cdot n\log n)$, where d is the feature size and n is the sample number; the complexity of the CBE algorithm is $O(K\cdot d\cdot N_e\cdot N_s)$, where K is the number of leaf nodes of the IC-Tree, and N_e and N_s are the number of initial explorers and auxiliary explorers. Therefore, the training time is theoretically linear to the feature size, which is in line with the empirical results. For execution, the time complexity is $O(|C|\cdot d)$, where $|C|$ is the number of extracted rules.

In summary, as a rule-based approach, our method does not have an operation with computational complexity higher than quadratic terms, indicating its reasonable efficiency for deployment. In the next section, we will also validate its complexity by empirical experiments.

VI. Framework Extensibility

The proposed rule extraction approach is not only effective for global explanation, but also exhibits remarkable extensibility to other forms of interpretability that are vital in practical security applications. By providing explicit and human-understandable characterizations of normal data boundaries, Ano2Rule offers a flexible foundation for supporting multiple interpretability tasks within a unified paradigm. This versatility enables the same set of extracted rules to serve as the basis for local explanations—such as feature attribution for individual predictions—as well as counterfactual reasoning, wherein actionable recourse or remediation steps can be systematically generated for anomalous samples.

A. Local Explanation via Rule-based Feature Attribution

Given an input sample $\boldsymbol{x}\in\mathbb{R}^d$, our framework provides instance-level interpretability by leveraging the explicit, axis-aligned boundaries of the rule C in the rule set $\mathcal{C}$ that covers $\boldsymbol{x}$. Suppose the matched rule C is defined as $C=$

$l_i \leq x_i \leq u_i, , \forall i = 1, \ldots, d$, where l_i and u_i denote the lower and upper bounds for the i-th feature. For each feature i, we define the attribution score w_i for $\boldsymbol{x}$ as

$$w_i = \begin{cases} 0, & \text{if } x_i \in [l_i, u_i] \\ x_i - u_i, & \text{if } x_i > u_i \\ l_i - x_i, & \text{if } x_i < l_i \end{cases} \tag{18}$$

This formulation quantifies the extent to which each feature of $\boldsymbol{x}$ deviates from the corresponding boundary of C. If x_i lies within the bounds $[l_i, u_i]$, it is considered to contribute nothing to the anomaly for this rule. If x_i exceeds the upper or lower bound, the distance to the closest boundary is taken as its contribution to the model's decision.

For samples classified as normal (i.e., $x_i \in [l_i, u_i]$ for all i), we further characterize the feature-wise "tightness" of C by considering the interval width: a smaller width $u_i - l_i$ implies greater sensitivity of the model's decision to changes in x_i. One may define a potential importance score for such features as $w_i = 1/(u_i - l_i)$.

In summary, this rule-based attribution provides a local explanation of how each feature of $\boldsymbol{x}$ influences the model's prediction, allowing analysts to clearly identify which feature deviations are most responsible for the detection result.

B. Counterfactual Explanation via Rule-based Projection

Given an anomalous sample $\boldsymbol{x} \in \mathbb{R}^d$ not covered by any rule in the set $\mathcal{C}$, our framework constructs counterfactual explanations by minimally modifying $\boldsymbol{x}$ so that the altered sample $\boldsymbol{x}^{\text{cf}}$ falls strictly inside the boundary of at least one rule $C \in \mathcal{C}$. To ensure that the counterfactual sample is robustly accepted as normal (i.e., not exactly on the boundary), we introduce a small margin $\boldsymbol{\delta}$ for each feature, proportional to the width of the rule's interval.

Formally, for a given rule $C = l_i \leq x_i \leq u_i, \ \forall i = 1, \ldots, d$, we define $\delta_i = \gamma \cdot (u_i - l_i)$ where $\gamma > 0$ is a margin factor, e.g., $\gamma = 0.05$. The counterfactual for each feature is then computed as:

$$x_i^{\text{cf}} = \begin{cases} l_i + \delta_i, & \text{if } x_i < l_i \\ u_i - \delta_i, & \text{if } x_i > u_i \\ x_i, & \text{if } x_i \in [l_i, u_i] \end{cases} \tag{19}$$

To select the most plausible counterfactual, we evaluate candidates for all $C \in \mathcal{C}$ and minimize the following cost function:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{x}^{\text{cf}}) = \alpha \cdot |\boldsymbol{x} - \boldsymbol{x}^{\text{cf}}|_1 + (1 - \alpha) \cdot S(\boldsymbol{x}, \boldsymbol{x}^{\text{cf}}) \tag{20}$$

where $|\cdot|_1$ denotes the ℓ_1 distance, and $S(\boldsymbol{x}, \boldsymbol{x}^{\text{cf}})$ is the sparsity term defined as the number of features for which $x_i \neq x_i^{\text{cf}}$. Here, α is a weighting coefficient balancing proximity and sparsity and we set $\alpha = 0.5$.

The optimal counterfactual explanation is then given by

$$\boldsymbol{x}^{\text{cf}*} = \arg \min_{\boldsymbol{x}^{\text{cf}} \in \mathcal{P}(\mathcal{C}, \boldsymbol{x})} \mathcal{L}(\boldsymbol{x}, \boldsymbol{x}^{\text{cf}}) \tag{21}$$

where $\mathcal{P}(\mathcal{C}, \boldsymbol{x})$ is the set of all margin-adjusted projections of $\boldsymbol{x}$ into rules $C \in \mathcal{C}$ that yield a normal prediction. This approach ensures that the generated counterfactuals are both actionable (few and small changes) and robust (strictly inside the rule), providing clear, operational guidance for remediation in security applications.

TABLE III: Summary of datasets for network intrusion detection and AUC of trained models.

Dataset	#Features	#Attack	AE	VAE	OCSVM	iForest
CIC-IDS2017	80	29.55%	0.9921	0.9901	0.9967	0.9879
CSE-CIC-IDS2018	80	22.62%	0.9906	0.9767	0.9901	0.9734
TON-IoT	30	74.29%	0.9998	0.9998	0.9993	0.9877
CIC-IOT2023	47	41.49%	0.9756	0.9861	0.9778	0.8955

VII. Evaluation

This section evaluates the performance and effectiveness of the rule extraction method proposed for unsupervised anomaly detection models. Through experiments on four datasets, we verify the advantages of this method in terms of rule extraction accuracy, robustness, and interpretability.

A. Experimental Setup

In this study, we employ four widely used unsupervised anomaly detection models as black-box models, including Autoencoders (AE) [1], Variational Autoencoders (VAE) [58], One-Class Support Vector Machines (OCSVM) [59], and Isolation Forests (iForest) [8]. The experiments are conducted using four benchmark network traffic datasets: CIC-IDS2017, CSE-CIC-IDS2018[1] [61], TON-IoT [62], and CIC-IoT [63]. These datasets cover different feature dimensions and attack-to-normal traffic ratios, facilitating a comprehensive evaluation of the framework across multiple anomaly detection scenarios. Each dataset is represented in tabular format, with each row corresponding to a network traffic record and each column representing a statistical feature, such as the mean packet size and inter-arrival time intervals. Additionally, each dataset includes multiple attack types; for instance, the CIC-IDS2017 dataset contains attacks such as DDoS, DoS, and others.

The datasets are randomly divided into training, validation, and testing sets with a 6:2:2 ratio. Only normal traffic data is used to train the anomaly detection models and calibrate their hyperparameters. The dataset descriptions, along with the AUC (Area Under the Curve) scores for each model, are detailed in Table III. This comprehensive evaluation approach enables us to assess the framework's performance in detecting anomalies under varying feature characteristics and attack prevalence rates, thereby providing robust validation of its capabilities.

Baselines. We employ five prior explanation methods as baselines: 1) We use [48] that extracts rules from unsupervised anomaly detection (UAD); 2) For other global methods, we use the estimated greedy decision tree (EGDT) proposed by [19], and Trustee [20] that specifically explains security applications; 3) We also consider one method LIME [11] that can use a Submodular Pick algorithm to aggregate local explanations into global explanations, and a knowledge distillation (KD) method [55] that globally converts a black-box model to a

[1]We use the corrected versions of these two datasets as released by recent studies [60] to address known labeling issues.

TABLE IV: Performance of rule extraction on different datasets.

CIC-IDS2017 dataset

Method	AE				VAE				OCSVM				iForest			
	FD	RB	TPR	TNR	FD	RB	TPR	TNR	FD	RB	TPR	TNR	FD	RB	TPR	TNR
UAD	0.1325	0.4991	0.0003	0.9792	0.1438	0.4839	0.022	**0.9988**	0.0725	0.5000	0.00	1.00	0.1262	0.5000	0.0	**1.00**
EGDT	0.533	**1.00**	0.4354	0.9947	0.1437	**1.00**	0.022	0.9961	0.9189	0.9994	0.9306	0.838	0.9729	0.9996	0.9417	0.9189
Trustee	0.4871	0.6412	0.3844	0.9981	0.1552	0.9857	0.0152	**0.9988**	0.539	0.6108	**1.00**	**1.00**	0.4543	0.5801	0.9795	0.4486
LIME	0.6918	0.9999	0.7889	0.0014	0.8232	**1.00**	0.9329	0.001	0.068	0.9999	0.0777	0.0241	0.8910	**0.9998**	0.8246	0.9913
KD	0.5776	0.9989	0.4792	**0.9998**	0.2010	0.9817	0.1016	0.9993	0.3620	**1.00**	0.3102	0.9995	0.1262	0.7016	0.00	**1.00**
Ano2Rule	**0.9835**	**1.00**	**0.9457**	0.9915	**0.9620**	0.9993	**0.9610**	0.9944	**0.9275**	**1.00**	**1.00**	**1.00**	**1.00**	0.9949	**0.9968**	0.9843

CSE-CIC-IDS2018 dataset

Method	AE				VAE				OCSVM				iForest			
	FD	RB	TPR	TNR	FD	RB	TPR	TNR	FD	RB	TPR	TNR	FD	RB	TPR	TNR
UAD	0.3796	0.3077	0.0004	0.7418	0.2697	0.2930	0.1490	0.4857	0.6051	0.3069	0.3004	0.9876	0.6811	0.4035	0.3539	0.9724
EGDT	0.5821	**1.00**	0.1432	0.9801	0.2197	0.9989	0.2308	0.9554	0.5106	**1.00**	**1.00**	0.9546	0.9546	0.7813	0.9888	0.8971
Trustee	0.5157	0.9006	0.1901	0.9857	0.3642	0.9752	0.0124	0.9636	0.3616	0.5955	**1.00**	**1.00**	0.4241	0.4700	0.9641	0.5162
LIME	0.5838	0.9997	0.7681	0.0255	0.6814	**1.00**	0.9402	0.0213	0.0560	**1.00**	0.9999	0.0186	0.8903	**1.00**	0.9884	0.8745
KD	0.5074	0.9999	0.3562	**0.9979**	0.4234	0.9989	0.1086	**0.9925**	0.3180	0.9967	0.4308	0.1510	0.3596	0.6834	0.0000	**1.00**
Ano2Rule	**0.9954**	0.9997	**0.9998**	0.9774	**0.8962**	0.9985	**0.9997**	0.8268	**0.9929**	0.9997	0.9983	0.9753	**0.9947**	0.9291	**0.9988**	0.9583

TON-IoT dataset

Method	AE				VAE				OCSVM				iForest			
	FD	RB	TPR	TNR	FD	RB	TPR	TNR	FD	RB	TPR	TNR	FD	RB	TPR	TNR
UAD	0.1499	0.015	0.0258	0.908	0.2157	0.4010	0.1863	0.7787	0.0489	0.5000	0.00	**1.00**	0.0674	0.5000	0.00	**1.00**
EGDT	0.9750	**1.00**	0.9739	0.9943	0.7660	**1.00**	0.7538	0.9948	0.8139	0.9997	0.8051	0.9759	0.6345	0.9226	0.6247	0.9475
Trustee	0.4774	0.5722	0.4502	0.9971	0.3807	0.6689	0.3484	0.9975	0.7942	0.8430	**1.00**	**1.00**	0.7476	0.8145	0.9824	0.1943
LIME	0.6971	0.9999	0.7939	0.0027	0.8289	**1.00**	0.9379	0.0015	0.0687	0.9999	0.0787	0.0231	0.8963	**0.9998**	0.8296	0.9918
KD	0.0821	**1.00**	0.0341	**0.9987**	0.0591	0.9997	0.0099	**0.9980**	0.0494	**1.00**	0.0005	0.9994	0.0674	0.9955	0.00	**1.00**
Ano2Rule	**0.9996**	**1.00**	**1.00**	0.9845	**0.9995**	**1.00**	**1.00**	0.9831	**0.9511**	**1.00**	**1.00**	0.9881	**1.00**	0.9890	**1.00**	0.9715

CIC-IoT dataset

Method	AE				VAE				OCSVM				iForest			
	FD	RB	TPR	TNR	FD	RB	TPR	TNR	FD	RB	TPR	TNR	FD	RB	TPR	TNR
UAD	0.889	0.00	0.00	**1.00**	0.3644	0.4834	0.8913	**0.9988**	0.877	0.2583	0.1535	0.9505	0.829	0.0093	0.005	0.9405
EGDT	0.8975	**1.00**	0.0941	0.9911	0.9495	0.939	0.5891	0.9928	0.912	**1.00**	0.3168	0.9911	0.8905	0.9985	0.7475	0.8187
Trustee	0.8865	0.3673	0.0891	0.9828	**0.9595**	0.9119	0.7178	0.9922	0.0705	0.9398	0.9967	0.4356	0.105	0.0472	0.3861	0.1246
LIME	0.1015	0.1095	**0.995**	0.0011	0.109	0.111	**1.00**	0.0089	0.1585	0.1585	**1.00**	0.064	0.1145	0.2005	0.8465	0.0323
KD	0.881	0.999	0.00	0.9889	0.1575	**1.00**	0.4653	0.8826	0.2555	**1.00**	0.1139	0.7286	0.7865	**0.999**	0.2476	**0.9855**
Ano2Rule	**0.9265**	0.9985	0.8168	0.9277	0.931	0.993	0.7772	0.9461	**0.967**	0.998	0.8861	**0.9861**	**0.888**	0.994	**0.9851**	0.8315

self-explained decision tree. These methods, like ours, can only access normal data to extract explanations.

Metrics. We refer to the metrics outlined in [22] to evaluate rule extraction methods. The following key evaluation metrics are presented in this section: *Fidelity (FD)* refers to the proportion of samples where the predictions of the original model and the surrogate model are consistent. This metric reflects the trustworthiness of the explanation. *Robustness (RB)* measures the ability of the surrogate model to maintain the same prediction as the original model when the input undergoes small perturbations, indicating the method's stability. *True Positive Rate (TPR)* and *True Negative Rate (TNR)* evaluate the method's ability to detect positive and negative class samples, respectively. These metrics indicate whether a method meets the requirements for global interpretation and helps avoid alarm fatigue caused by false positives in highly imbalanced security applications [64].

B. Quality of Rule Extraction

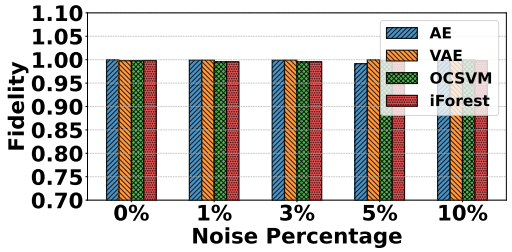
In our comprehensive performance evaluation (as shown in Table IV), Ano2Rule demonstrates superior performance across various anomaly detection models and datasets. First, it consistently achieves high levels of True Positive Rate (TPR) and True Negative Rate (TNR). For example, on the TON-IoT dataset, all detection models reach a TPR of 1.00, while the lowest TNR is as high as 0.9715. This highlights that our method not only accurately detects anomalous data but also effectively identifies normal data, significantly reducing false positives and false negatives. Moreover, Ano2Rule exhibits outstanding performance in terms of Fidelity (FD) and Robustness (RB). The Fidelity scores across all datasets exceed 0.95, with more than half of them surpassing 0.99. This indicates that our method can accurately replicate the predictions of the black-box models, ensuring the correctness of global explanations. The RB scores range between 0.9890 and 1.00 across all datasets, demonstrating that the method maintains stable performance even under data noise and variations, thereby ensuring its reliability in real-world deployments.

The baseline methods typically exhibit considerable variability in metrics across different datasets. In contrast, on the CIC-IDS2017 dataset, Ano2Rule improves Fidelity by approximately 0.45 (from 0.533 to 0.9835) and TNR by about

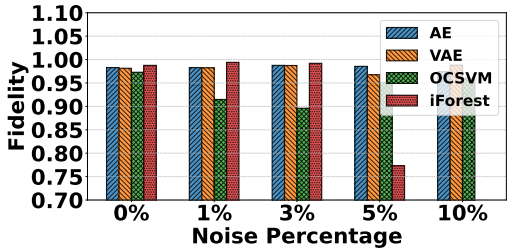
0.16 (from 0.9947 to 0.9915) compared to EGDT. On the TON-IoT dataset, Ano2Rule achieves perfect TPR and TNR values of 1.00, substantially outperforming LIME and KD. These results demonstrate that our method not only excels across different datasets but also offers highly reliable explanations for security applications. By efficiently extracting rules, Ano2Rule provides effective global interpretations for unsupervised black-box models, offering security analysts precise and transparent anomaly detection support.

Notably, although Ano2Rule may not be the top performer on every individual metric, it delivers highly consistent and robust results: for example, Ano2Rule achieves a score above 0.95 on 50 out of 64 (78.1%) evaluation settings, covering all datasets, black-box models, and metrics. In contrast, some baseline methods exhibit larger fluctuations, with high scores only on certain datasets but significant drops elsewhere. This level of stability and balanced performance underlines the practical value of Ano2Rule in real-world applications, where dependable interpretability across a variety of conditions is often more important than isolated peak results.

Considering that obtaining a "clean" training set requires huge manual effort in reality [65], we also assess the efficacy of Ano2Rule under varying percentages of "noisy" data. We evaluate the fidelity of extracted rules using two approaches for the injection of noisy data: 1) random noise; 2) mislabeled data from other classes, i.e., attack data. The results are shown in Figure3. We find that the impact of the noisy data proportion is not significant: 36 of 40 fidelity scores in the table preserve over 0.95, and the variation of fidelity scores is not obvious with the increase of noisy data for most of the models. This shows that our rule extraction method can retain similar performance to the black-box model that it extracts from. Nonetheless, the results of iForest also reveal that a sufficiently large proportion of noisy data may cause a certain negative impact on the rule extraction for certain models. Overall, our method achieves high-quality rule extraction for anomaly detection models by balancing sensitivity and specificity. This capability makes it a highly reliable solution in noisy environments.



(a) FD under Mislabeled Noise



(b) FD under Random Noise

Fig. 3: Fidelity of extracted rules under varying percentages of noisy training data.

C. Understanding Model Decisions

To demonstrate that the rules obtained by our method are in line with human understanding, we use OCSVM as an example of black-box models to provide several explanations. We extract rules from a well-trained model and use these rules to predict four typical types of attack data, including Distributed Denial-of-Service (DDoS) attacks, XSS attacks, password attacks, and ransomware attacks. Table V presents the features of the rules extracted from normal data that cannot be matched by the attack data and explains how humans can interpret the model's decisions. Such explanation results are obtained by the following steps:

1) For a reported anomaly $\boldsymbol{x}$, we use the IC-Tree to pinpoint the rule of normality that judges $\boldsymbol{x}$ as anomalous. It is realized by inputting $\boldsymbol{x}$ into the tree and recursively finding the leaf node. The rule is denoted by $C_{\boldsymbol{x}}$.
2) For each constraint of features in $C_{\boldsymbol{x}}$, we compare the corresponding feature value of $\boldsymbol{x}$ with the constraint.
3) For the feature values outside the range of the constraints, the Rules of Normality, Feature Values, and Feature Meaning (i.e., the three columns in Table V) along with the raw data sample will be sent as the explanation to the security expert for further analysis.
4) The security expert will analyze the data sample to determine the type of attack (i.e., the Attack column in Table V) and give her/his understanding of the important attributes that make her/him identify the attack (i.e., the Human Understanding column in Table V).
5) If there is a huge gap between the provided explanation and expert understanding, it indicates that the anomaly detector may not be trustworthy.

For instance, the data of DDoS attacks fails to match the rules in three feature dimensions: the mean of packet sizes, the mean of packet inter-arrival time, and the duration of a connection. Specifically, the rule "$ps_{mean} > 101.68$" is unmet, as the mean packet size in attack data is only 57.33, indicating that the packet size in DDoS attacks is significantly smaller. Similarly, the rule "$iat_{mean} > 0.063$" is unmet, with the attack value being merely 0.00063, demonstrating extremely short inter-arrival times between packets. The rule "$dur > 12.61$" is also unmet, as the attack value is only 0.00126, reflecting the short connection durations characteristic of DDoS attacks. Such results are easy to interpret: the purpose of DDoS attacks is to overwhelm the victim's resources, and attackers achieve this by consuming resources asymmetrically (i.e., using small packets), sending packets at an extremely high rate (i.e., low inter-arrival time), and establishing as many useless connections as possible (i.e., short connection durations). These explanations align with how humans recognize DDoS attack behaviors.

These results clearly demonstrate that our extracted rules align with human understanding of attack patterns, showing that our method can provide precise, human-readable insights

TABLE V: Examples of explanation on four types of attacks.

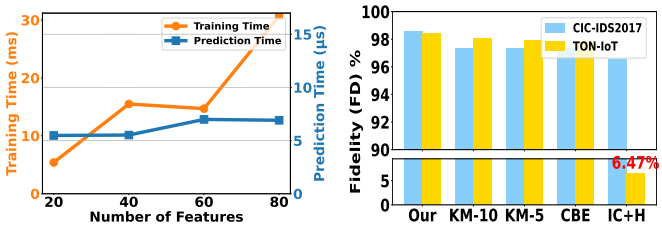
Attack	Rules of Normality	Attack Value	Feature Meaning	Human Understanding
DDoS	ps_mean $>$ 101.68 iat_mean $>$ 0.063 dur $>$ 12.61	57.33 0.00063 0.00126	Mean of IP packet sizes Mean of inter-arrival time (IAT) Duration of a connection	DDoS attacks flood the victim with small packets at high rates, consuming resources.
XSS	ps_mean $\leq$ 185.30 ps_var $\leq$ 107,333.64 ps_bwd_var $\leq$ 87,446.91	185.30 87,446.91 87,446.91	Mean of IP packet sizes Variance of IP packet sizes Variance of backward IP packet sizes	XSS attacks inject malicious scripts into web pages, affecting user experience.
Password	ps_mean $\leq$ 185.30 iat_bwd_var $>$ 0.0970	185.30 0.0970	Mean of IP packet sizes Variance of backward packet IAT	Password attacks attempt to brute-force passwords through repeated login attempts.
Ransomware	ps_mean $\leq$ 186.62 ps_var $>$ 107,775.82 ps_bwd_mean $>$ 194.58	186.62 107,775.82 194.58	Mean of IP packet sizes Variance of IP packet sizes Mean of backward IP packet sizes	Ransomware attacks encrypt data and demand ransom, with high packet size variability.

into black-box anomaly detection models. Besides, we note that the gap between the provided explanation and human understanding may not only occur when the anomaly detector makes erroneous decisions but also when the anomaly detector correctly detects an attack sample by unreasonable features, which is called "spurious correlation" or "shortcut learning" [66]. For example, the most obvious attribute of a DoS attack is typically its high rate of forwarding packets in order to overwhelm the victim, while the provided explanation might suggest that IP address is an important feature. Specifically, unreliable decisions could be made due to inappropriate testbed settings during the collection of training data, such as the DoS attack being launched from one separate host address while all other normal traffic is from other host addresses. Therefore, our explanation can also verify whether the detectors are correctly trained before they are deployed.

D. Ablation Study

To evaluate the contributions of each component in Ano2Rule, including the IC-Tree and the CBE algorithm, we conducted ablation experiments with the following configurations: 1) replacing the IC-Tree with the K-Means clustering algorithm; 2) using only the CBE algorithm; and 3) replacing the CBE algorithm with directly using hypercubes as rules. As shown in Figure 4b, our method (IC-Tree + CBE) outperforms the other methods in terms of fidelity (FD) on both datasets.

For instance, in the comparison between using only CBE and the combination of IC-Tree + CBE, the fidelity on the CIC-IDS2017 dataset increased from 0.9735 to 0.9856, an improvement of 1.24%. On the TON-IoT dataset, the fidelity improved from 0.9784 to 0.9840, an increase of 0.57%. These results indicate that incorporating IC-Tree significantly enhances the model's fidelity across different datasets. Furthermore, in the ablation experiments, although using K-Means (e.g., $k = 10$ or $k = 5$) achieved comparable fidelity results, its clustering results are difficult to express through axis-aligned rules, leading to poor interpretability and deployability. In contrast, our method (IC-Tree + CBE) not only achieves higher fidelity but also enhances the model's interpretability and deployment potential through clear rule representation. In summary, these experimental results validate the contributions of each component and demonstrate the advantages of Ano2Rule.



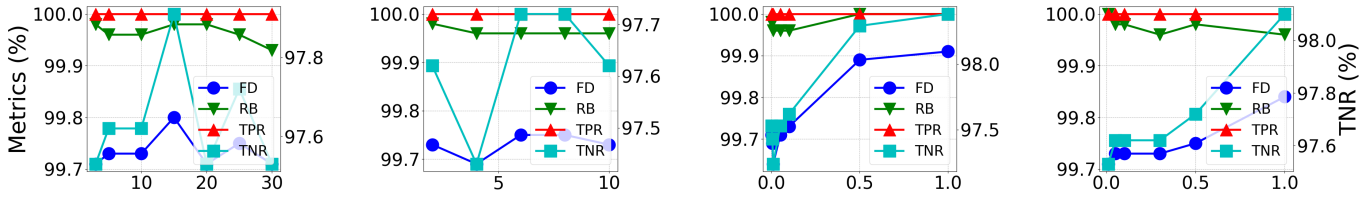
(a) Average training and prediction time (b) Ablation study results

Fig. 4: (a) Average training and prediction time per sample for different feature sizes. (b) Ablation study evaluating the contribution of each component to overall performance.

E. Computational Cost

We also evaluate the computational cost of Ano2Rule with respect to training and prediction. Since the CIC-IDS2017 dataset has 80 features in total, we train the model using the first 20, 40, 60, and 80 features of 4000 samples to investigate the influence of feature sizes. The results are shown in Figure 4a, which demonstrates the average training and prediction time of our method. The experimental results indicate that training time increases linearly with the number of features. This is because Ano2Rule adopts a feature-by-feature strategy to explore the decision boundary of the model. As data complexity increases, each additional feature expands the data space, forcing the model to learn a higher-dimensional representation, thus extending the training time. Nevertheless, the training time is around 1 minute, which is acceptable and practical for large-scale training.

Regarding prediction time, Ano2Rule is highly efficient, with each inference taking only microseconds. This demonstrates that our method, as a rule-based approach, can achieve real-time execution for online applications. Furthermore, since the prediction phase relies on a set of predefined rules derived from the tree's decision paths and boundary delineations, the computational process remains lightweight, making it well-suited for real-time scenarios. It is worth noting that the runtime was measured entirely based on Python implementation. In practice, the prediction time could be further reduced with more efficient code implementations.



(a) Maximum tree depth (b) #Explorers (c) Sampling coefficient (d) Iteration stride

Fig. 5: Sensitivity experiments of hyperparameters.

F. Hyperparameter

We perform a sensitivity analysis of several hyperparameters on their influence on the rule extraction. We present four major hyperparameters in Figure 5, including the maximum depth τ of an IC-Tree, N_e number of explorers, the coefficient ρ of sampling, and the factor η that controls the stride of an iteration.

Maximum tree depth. A deeper IC-Tree creates more leaf nodes, which allows for finer decomposition of distributions, easing the difficulty of rule extraction. However, excessive depth can lead to overfitting, reducing generalization capability. As shown in Figure 5a, $\tau = 15$ provides the best balance between performance and complexity. This result underscores the importance of selecting an appropriate tree depth to avoid redundancy while ensuring sufficient resolution in feature space.

Number of Explorers. The number of explorers N_e in Beam Search determines the number of nodes evaluated per iteration. This parameter significantly affects algorithmic efficiency and accuracy. While a higher N_e helps explore multiple local optima, it can also introduce redundancy, as shown in Figure 5b. An optimal range of 6 to 8 explorers is recommended, ensuring robust performance without unnecessary computational overhead.

Coefficient of sampling. The sampling coefficient ρ determines the radius of sampling within a multivariate Gaussian distribution. Larger values of ρ enhance the CBE algorithm's ability to identify the decision boundary effectively by enabling a broader search radius, as illustrated in Figure 5c. This capability is crucial for ensuring accurate rule extraction, particularly in high-dimensional spaces where decision boundaries are complex.

Factor of iteration stride. The iteration stride factor η controls the step size in boundary exploration. Larger values of η improve convergence speed and rule quality by facilitating more significant adjustments during each iteration. Figure 5d demonstrates that higher values lead to better performance, suggesting that a balance between stride length and convergence stability is key for optimizing exploration.

G. Extensibility

To further demonstrate the versatility of Ano2Rule, we evaluate its effectiveness as a local and counterfactual explanation method, in comparison with specialized state-of-the-art approaches.

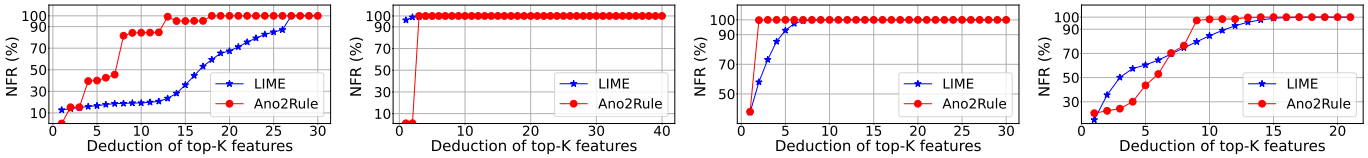
Local Explanation. To assess the effectiveness of Ano2Rule as a local explanation method, we compare its feature attribution capability with the widely used model-agnostic framework LIME [11]. We use the *Negative Flipping Rate (NFR)* as the evaluation metric, which quantifies the proportion of originally normal samples that are reclassified as anomalous by the black-box model after the top-K most important features—identified by the explanation method—are set to zero. A higher NFR at smaller K values indicates that the explanation method is more accurate in pinpointing the features most critical to the model's decision.

The comparative results are shown in Fig. 6. Across most benchmark datasets, Ano2Rule achieves a higher or comparable NFR with fewer features flipped, and the NFR curve rises more rapidly as K increases. For example, on the TON-IoT dataset, Ano2Rule achieves a 100% flipping rate with fewer than five features altered, whereas LIME requires more features to reach the same flipping ratio. On some datasets, the two methods perform similarly, reflecting that our global rule extraction can match the local interpretability provided by LIME.

Counterfactual Explanation. We further evaluate the counterfactual explanation capability of Ano2Rule by comparing it against DiCE [43], a widely adopted model-agnostic counterfactual generation approach. For each detected anomaly, both methods are used to generate multiple candidate counterfactual samples. We assess their performance using three metrics: *Validity* (the proportion of generated counterfactuals classified as normal by the original model), *Proximity* (the average feature-wise distance between each counterfactual and the original anomalous sample, indicating the extent of modification), and *Success Rate* (the proportion of anomalous samples for which at least one valid counterfactual is found).

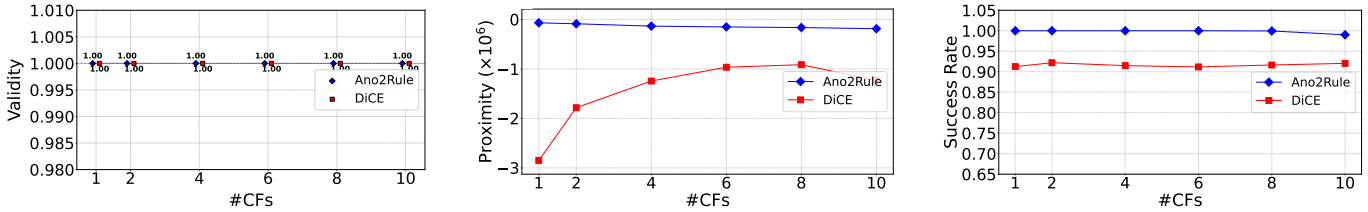
The comparative results are presented in Fig. 7. Across all evaluated datasets, Ano2Rule consistently achieves higher validity and success rates than DiCE, indicating that our rule-based approach is more effective at generating actionable recourse that leads to a change in model prediction. Notably, the proximity scores of Ano2Rule counterfactuals are lower on average, meaning that the modifications suggested by our method are not only more likely to succeed, but also require less perturbation to the original sample.

Overall, these results demonstrate that Ano2Rule, though originally designed for global explanation, can be readily extended to other types of explanation while preserving similar effectiveness to the specialized methods of these types. In addition, as it requires no iterative sampling or surrogate model training, our rule-based approach achieves higher explanation efficiency, making it more practical for large-scale or real-time

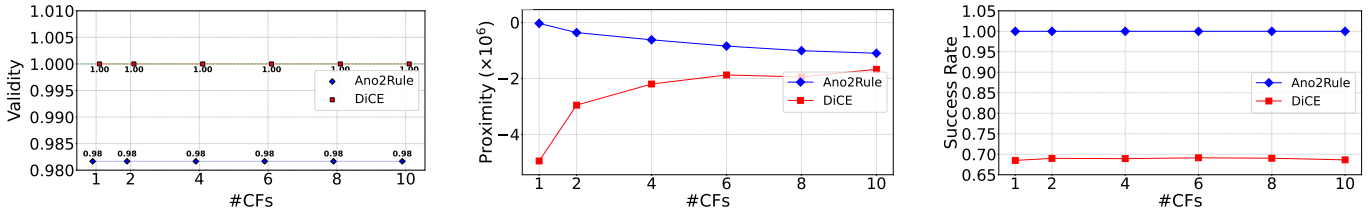


(a) CIC-IDS2017 (b) CSE-CIC-IDS2018 (c) TON-IoT (d) CIC-IoT

Fig. 6: Comparing Ano2Rule as a local explanation method to LIME.



(a) Validity (CIC-IDS2017) (b) Proximity (CIC-IDS2017) (c) Success Rate (CIC-IDS2017)



(d) Validity (CIC-IoT) (e) Proximity (CIC-IoT) (f) Success Rate (CIC-IoT)

Fig. 7: Comparing Ano2Rule as a counterfactual explanation method to DiCE.

applications.

VIII. Conclusion and Future Work

This paper presented a novel method called Ano2Rule for providing global explanations of black-box unsupervised anomaly detection models. The method combines the Internal Clustering Tree (IC-Tree) with a Boundary Exploration Algorithm (CBE) to efficiently extract rules and offer transparent explanations for anomaly detection. IC-Tree recursively partitions the feature space to capture the hierarchical structure of data distributions, while CBE further explores decision boundaries within each subspace to enhance detection accuracy. Ano2Rule not only ensures global interpretability but also provides fine-grained local detection capabilities, offering comprehensive and intuitive explanations of the anomaly detection model's reasoning process. Experimental results demonstrated that Ano2Rule outperforms existing approaches across multiple key metrics, including fidelity, robustness, true positive rate, and true negative rate, providing robust support for security applications.

Although Ano2Rule exhibits excellent global interpretability and high fidelity, there are still some limitations, which provide multiple avenues for future research. First, we are curious if it is possible to extend the method to other data modalities, such as raw image data. This may require integrating deep models with spatial awareness capabilities, such as Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), to extract high-level semantic features, thereby enhancing the method's adaptability for cross-domain applications. Second, to better balance interpretability and fitting capacity, we will explore alternative models and algorithms, such as more complex surrogate models, to accommodate the intricate shapes of decision boundaries in high-dimensional feature spaces. Lastly, given the method's generalizability, we will explore its application in other domains requiring interpretable anomaly detection, such as medical diagnostics, manufacturing monitoring, and criminal investigation analysis. These research directions will further enhance the applicability and practicality of Ano2Rule, providing robust support for its widespread use in various high-risk multi-domain scenarios.

References

[1] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Annual Network and Distributed System Security Symposium (NDSS)*, 2018.

[2] R. Tang, Z. Yang, Z. Li, W. Meng, H. Wang, Q. Li, Y. Sun, D. Pei, T. Wei, Y. Xu, and Y. Liu, "Zerowall: Detecting zero-day web attacks through encoder-decoder recurrent neural networks," in *IEEE Conference on Computer Communications (INFOCOM)*, 2020.

[3] C. Fu, Q. Li, M. Shen, and K. Xu, "Realtime robust malicious traffic detection via frequency domain analysis," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021.

[4] R. Perdisci, W. Lee, and N. Feamster, "Behavioral clustering of http-based malware and signature generation using malicious network traces," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2010.

[5] E. C. R. Shin, D. Song, and R. Moazzezi, "Recognizing functions in binaries with neural networks," in *USENIX Security Symposium*, 2015.

[6] M. Villarreal-Vasquez, G. Modelo-Howard, S. Dube, and B. Bhargava, "Hunting for insider threats using lstm-based anomaly detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 451–462, 2023.

[7] W. Fan, H.-J. Hong, J. Kim, S. Wuthier, M. Nakashima, X. Zhou, C.-H. Chow, and S.-Y. Chang, "Lightweight and identifier-oblivious engine for cryptocurrency networking anomaly detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1302–1318, 2023.

[8] Y. Dong, Q. Li, K. Wu, R. Li, D. Zhao, G. Tyson, J. Peng, Y. Jiang, S. Xia, and M. Xu, "Horuseye: Realtime iot malicious traffic detection framework with programmable switches," in *USENIX Security Symposium*, 2023.

[9] L. Xi, D. Miao, M. Li, R. Wang, H. Liu, and X. Huang, "Adaptive-correlation-aware unsupervised deep learning for anomaly detection in cyber-physical systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 2888–2899, 2024.

[10] A. Alsaedi, Z. Tari, R. Mahmud, N. Moustafa, A. Mahmood, and A. Anwar, "Usmd: Unsupervised misbehaviour detection for multi-sensor data," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 724–739, 2023.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[12] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[14] W. Guo, D. Mu, J. Xu, P. Su, G. Wang, and X. Xing, "Lemna: Explaining deep learning based security applications," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018.

[15] D. Han, Z. Wang, W. Chen, Y. Zhong, S. Wang, H. Zhang, J. Yang, X. Shi, and X. Yin, "Deepaid: Interpreting and improving deep learning-based anomaly detection in security applications," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021.

[16] netfilter project, "iptables," https://www.netfilter.org/projects/iptables/index.html, 2024.

[17] snort, "Snort ids," https://www.snort.org/, 2024.

[18] M. W. Craven and J. W. Shavlik, "Using sampling and queries to extract rules from trained neural networks," in *International Conference on Machine Learning (ICML)*, 1994.

[19] O. Bastani, C. Kim, and H. Bastani, "Interpreting blackbox models via model extraction," *CoRR*, vol. abs/1705.08504, 2017.

[20] A. S. Jacobs, R. Beltiukov, W. Willinger, R. A. Ferreira, A. Gupta, and L. Z. Granville, "Ai/ml for network security: The emperor has no clothes," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.

[21] J. H. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," *The Annals of Applied Statistics*, vol. 2, no. 3, 2008.

[22] G. Vilone, L. Rizzo, and L. Longo, "A comparative analysis of rule-based, model-agnostic methods for explainable artificial intelligence," in *Irish Conference on Artificial Intelligence and Cognitive Science*, 2020.

[23] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017.

[24] R. Li, Q. Li, J. Zhou, and Y. Jiang, "Adriot: An edge-assisted anomaly detection framework against iot-based network attacks," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 10 576–10 587, 2022.

[25] R. Li, Q. Li, Y. Huang, W. Zhang, P. Zhu, and Y. Jiang, "Iotensemble: Detection of botnet attacks on internet of things," in *European Symposium on Research in Computer Security (ESORICS)*, 2022.

[26] 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, 2001.

[27] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. A. Vandermeulen, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning (ICML)*, 2018.

[28] S. Itani, F. Lecron, and P. Fortemps, "A one-class classification decision tree based on kernel density estimation," *Appl. Soft Comput.*, vol. 91, p. 106250, 2020.

[29] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation forest," in *IEEE International Conference on Data Mining (ICDM)*, 2008.

[30] H. Xu, G. Pang, Y. Wang, and Y. Wang, "Deep isolation forest for anomaly detection," *CoRR*, vol. abs/2206.06602, 2022.

[31] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[32] J. Kauffmann, K.-R. Müller, and G. Montavon, "Towards explaining anomalies: A deep taylor decomposition of one-class models," *Pattern Recognition*, vol. 101, p. 107198, 2020.

[33] D. Kazhdan, B. Dimanov, M. Jamnik, and P. Liò, "MEME: generating RNN model explanations via model extraction," *CoRR*, vol. abs/2012.06954, 2020.

[34] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, and K. Müller, "Explainable deep one-class classification," in *International Conference on Learning Representations (ICLR)*, 2021.

[35] D. L. Aguilar, M. A. Medina-Pérez, O. Loyola-González, K.-K. R. Choo, and E. Bucheli-Susarrey, "Towards an interpretable autoencoder: A decision-tree-based autoencoder and its application in anomaly detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1048–1059, 2023.

[36] Y. Feng, J. Li, D. Sisodia, and P. Reiher, "On explainable and adaptable detection of distributed denial-of-service traffic," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 2211–2226, 2024.

[37] Z. Kong and K. Chaudhuri, "Understanding instance-based interpretability of variational auto-encoders," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[38] J. Crabbé and M. van der Schaar, "Label-free explainability for unsupervised models," in *International Conference on Machine Learning (ICML)*, 2022.

[39] O. Eberle, J. Büttner, F. Kräutli, K. Müller, M. Valleriani, and G. Montavon, "Building and interpreting deep similarity models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1149–1161, 2022.

[40] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning (ICML)*, 2017.

[41] J. Sipple, "Interpretable, multidimensional, multimodal anomaly detection with negative sampling for detection of device failure," in *International Conference on Machine Learning (ICML)*, 2020.

[42] S. Wachter, B. D. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *CoRR*, vol. abs/1711.00399, 2017.

[43] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona (FAT)*, 2020, pp. 607–617.

[44] M. Bajaj, L. Chu, Z. Y. Xue, J. Pei, L. Wang, P. C. Lam, and Y. Zhang, "Robust counterfactual explanations on graph neural networks," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021, pp. 5644–5655.

[45] K. H. Tran, A. Ghazimatin, and R. S. Roy, "Counterfactual explanations for neural recommenders," in *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2021, pp. 1627–1631.

[46] B. A. Cumi-Guzman, A. D. Espinosa-Chim, M. G. Orozco-del-Castillo, and J. A. Recio-García, "Counterfactual explanation of a classification model for detecting SQL injection attacks," in *Workshops at the 32nd International Conference on Case-Based Reasoning (ICCBR-WS)*, 2024, pp. 49–64.

[47] R. McGrath, L. Costabello, C. L. Van, P. Sweeney, F. Kamiab, Z. Shen, and F. Lécué, "Interpretable credit application predictions with counterfactual explanations," *CoRR*, vol. abs/1811.05245, 2018.

[48] A. Barbado, O. Corcho, and R. Benjamins, "Rule extraction in unsupervised anomaly detection for model explainability: Application to oneclass svm," *Expert Systems with Applications*, vol. 189, p. 116100, 2022.

[49] D. Han, Z. Wang, R. Feng, M. Jin, W. Chen, K. Wang, S. Wang, J. Yang, X. Shi, X. Yin, and Y. Liu, "Rules refine the riddle: Global explanation for deep learning-based anomaly detection in security applications," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024, pp. 4509–4523.

[50] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model," *CoRR*, vol. abs/1511.01644, 2015.

[51] I. van der Linden, H. Haned, and E. Kanoulas, "Global aggregations of local explanations for black box models," *CoRR*, vol. abs/1907.03039, 2019.

[52] M. Setzu, R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti, "Glocalx - from local to global explanations of black box AI models," *Artif. Intell.*, vol. 294, p. 103457, 2021.

[53] Q. Li, R. Cummings, and Y. Mintz, "Optimal local explainer aggregation for interpretable prediction," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[54] N. Frosst and G. E. Hinton, "Distilling a neural network into a soft decision tree," *CoRR*, vol. abs/1711.09784, 2017.

[55] Y. Li, J. Bai, J. Li, X. Yang, Y. Jiang, and S. Xia, "Rectified decision trees: Exploring the landscape of interpretable and effective machine learning," *CoRR*, vol. abs/2008.09413, 2020.
[56] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, 1984.
[57] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICLR)*, 2015.
[58] X. Xu, J. Li, Y. Yang, and F. Shen, "Toward effective intrusion detection using log-cosh conditional variational autoencoder," *IEEE Internet of Things Journal*, pp. 6187–6196, 2021.
[59] A. Binbusayyis and T. Vaiyapuri, "Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM," *Appl. Intell.*, vol. 51, no. 10, pp. 7094–7108, 2021.
[60] L. Liu, G. Engelen, T. M. Lynar, D. Essam, and W. Joosen, "Error prevalence in NIDS datasets: A case study on CIC-IDS-2017 and CSE-CIC-IDS-2018," in *10th IEEE Conference on Communications and Network Security (CNS)*, 2022, pp. 254–262.
[61] ——, "Error prevalence in NIDS datasets: A case study on CIC-IDS-2017 and CSE-CIC-IDS-2018," in *IEEE Conference on Communications and Network Security (CNS)*, 2022.
[62] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 485–496, 2022.
[63] S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong, and A. A. Ghorbani, "Towards the development of a realistic multidimensional iot profiling dataset," in *19th Annual International Conference on Privacy, Security & Trust (PST)*, 2022.
[64] B. A. AlAhmadi, L. Axon, and I. Martinovic, "99% false positives: A qualitative study of SOC analysts' perspectives on security alarms," in *USENIX Security Symposium*, 2022.
[65] G. Apruzzese, P. Laskov, and A. Tastemirova, "Sok: The impact of unlabelled data in cyberthreat detection," in *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2022.
[66] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *USENIX Security Symposium*, 2022.