KNNIFE: a data-informed feature importance for Isolation Forest and its extensions

Riccardo De Vidi[1], Francesco Borsatti[1], Valentina Zaccaria[1], Davide Sartor[2], Gian Antonio Susto[1†]

***Abstract*— Isolation Forest and its variants are widely deployed for unsupervised anomaly detection in industrial environments due to their computational efficiency and scalability, often outperforming deep learning approaches on small/medium-sized tabular data. However, these methods typically flag anomalous instances without explaining contributing factors, limiting support for root cause analysis. We introduce K-Nearest Neighbors Isolation Forest Explanations (KNNIFE), a novel interpretability method that provides both global and local feature importance scores for any Isolation Forest variant. By integrating dataset-dependent information into the attribution process, KNNIFE outperforms existing Isolation Forest-specific explanation techniques on real-world and simulated industrial data, enhancing support for industrial root cause analysis and actionability.**

I. INTRODUCTION

Anomaly detection is a fundamental problem in data-driven industrial systems, with applications spanning fault detection in manufacturing, predictive maintenance of rotating machinery, quality control along production lines, and cybersecurity of industrial networks [1], [2], [3]. The core objective is to identify instances that deviate significantly from expected operational behavior, enabling timely interventions that prevent costly downtime, safety hazards, or product defects. Among the many classes of anomaly detection algorithms, ensemble methods based on the Isolation Forest (IF) [4] paradigm have gained widespread adoption in industry [5], [6]. Their appeal stems from a favorable combination of being unsupervised, i.e., they don't require labeled examples, strong detection performance on tabular data, often competitive with or superior to deep neural network approaches, linear time complexity and low memory footprint [7]. These properties make IF-based methods particularly interesting for deployment on resource-constrained devices operating at the edge of production lines, where inference latency and energy budgets are tightly constrained. Several extensions of the original IF algorithm have been proposed to address its geometric biases, including Extended Isolation Forest (EIF) [8], which replaces axis-aligned splits with oblique hyperplane partitions, and Deep Isolation Forest (DIF) [9], which maps data through random representations before applying the isolation procedure.

Despite their effectiveness, IF-based methods lack inherent interpretability. Their tree-based ensemble structure, based on randomized splits and, in the case of extensions, complex partition geometries, makes it difficult for domain experts to understand why a particular data point has been flagged as anomalous. In industrial settings, the opacity of these methods creates a major barrier. Plant operators and engineers need to understand which sensor readings or process variables triggered an anomaly detection in order to diagnose root causes and take proper corrective actions. Interpretability is therefore a prerequisite for the trustworthy deployment of anomaly detection in safety-critical environments.

This interpretability gap has been addressed by model-specific feature importance methods, which assign each input variable a score representing its contribution to the anomaly score. Depth-based Isolation Forest Feature Importance (DIFFI) [10], designed for standard IF, assigns the importance of a feature considering two properties of the splits in which it is involved: whether the split is imbalanced and the anomaly score of points traversing that node. Extended Isolation Forest Feature Importance (ExIFFI) [11] and FUBIFFI [12] extend this idea to EIF and any isolation-based anomaly detector, respectively. All these approaches are locally (within a node split) insensitive to the data distribution, explaining only the model inner structure. However, we argue that incorporating properties of the data distribution can better identify the features that actually drive the anomalousness of a data point.

The key idea is to replace, in the computation of the importance scores, the vectors used in ExIFFI, which depend only on the split geometry, with *data-induced vectors* that capture the features that truly separate instances across each split. Using K-Nearest Neighbors (KNN) between points on opposite sides of a split-induced partition, KNNIFE produces global and local feature importance scores that are applicable to *any* extension of the Isolation Forest algorithm, regardless of whether its splits are axis-aligned, linear, or nonlinear.

Importantly, KNNIFE preserves the inference computational efficiency that makes IF-based methods effective for industrial edge deployment. The data-induced importance vectors are computed from the training data stored at each tree node and do not require access to the decision boundary function itself, keeping the additional interpretability overhead modest and predictable.

The main contributions of this work are as follows:

- We introduce KNNIFE, a post-hoc model-specific interpretability method that provides global and local feature importance for any IF-based anomaly detection model, that explicitly accounts for data distribution.
- We evaluated KNNIFE on real-world anomaly detection benchmarks, demonstrating performance comparable to or superior to DIFFI and ExIFFI on standard IF and

[1] University of Padova.
[2] University of Massachusetts Amherst.
[†] Corresponding author: gianantonio.susto@unipd.it

EIF models, while additionally enabling interpretation of nonlinear extensions.
- We validate KNNIFE on two industrial datasets with known root-cause ground truth, showing that the resulting feature importance rankings are aligned with the true anomaly causes.

The remainder of this paper is organized as follows. Section II presents IF and reviews related work on interpretability for anomaly detection. Section III presents the KNNIFE methodology. Section IV describes the experimental setup and results[1]. Section V draws conclusions.

II. RELATED WORK

A. Isolation Forest and Extensions

Let $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a dataset of n points in $\mathbb{R}^d$. An Isolation Forest $\mathcal{F} = \{t_1, \ldots, t_T\}$ consists of T isolation trees (iTrees), each constructed from a bootstrap sample $\mathcal{D}_j$ of size ψ drawn uniformly with replacement from $\mathcal{D}$.

Each iTree is a binary tree built by recursive random partitioning. At each internal node v, a feature index $f(v)$ is selected uniformly at random, and a split value $s(v)$ is drawn uniformly from the range of feature $f(v)$ in the data $\mathcal{D}_v$ associated with v. The data is then partitioned as:

$$\mathcal{D}_{v_l} = \{\mathbf{x} \in \mathcal{D}_v \mid x_{f(v)} < s(v)\}, \tag{1}$$

$$\mathcal{D}_{v_r} = \{\mathbf{x} \in \mathcal{D}_v \mid x_{f(v)} \geq s(v)\}. \tag{2}$$

Recursion terminates when the maximum tree depth $d_{\max} = \lceil \log_2 \psi \rceil$ is reached or a node contains a single instance.

The anomaly score of a point $\mathbf{x}$ is defined as:

$$a(\mathbf{x}) = 2^{-\frac{\mathbb{E}[d_j(\mathbf{x})]}{c(n)}}, \tag{3}$$

where the numerator $\mathbb{E}[d_j(\mathbf{x})] = \frac{1}{T}\sum_{j=1}^{T} d_j(\mathbf{x})$ is the average path length of $\mathbf{x}$ across all trees, with $d_j(x) := |\text{path}(\mathbf{x}|t_j)|$, and the denominator $c(n) = 2H(n-1) - 2(n-1)/n$ is the average path length of an unsuccessful search in a binary search tree, with $H(k) \approx \ln(k) + \gamma$ denoting the k-th harmonic number. Higher scores indicate the sample is more anomalous. The score can be converted into a binary label using a threshold τ, i.e., $y = (a(\mathbf{x}) \geq \tau)$.

In extensions such as EIF, the axis-aligned split is replaced by an oblique hyperplane split: at node v, a normal vector $\mathbf{u}(v)$ and an intercept point $\mathbf{p}(v)$ are generated, and data is partitioned according to $(\mathbf{x} - \mathbf{p}(v)) \cdot \mathbf{u}(v) \leq 0$. More generally, any extension can be characterized by an arbitrary splitting function $f_{q(v)} : \mathbb{R}^d \to \mathbb{R}$ and a threshold $s(v)$, yielding $\mathcal{D}_{v_l} = \{\mathbf{x} \in \mathcal{D}_v \mid f_{q(v)}(\mathbf{x}) \leq s(v)\}$ and $\mathcal{D}_{v_r} = \mathcal{D}_v \setminus \mathcal{D}_{v_l}$. In this perspective, DIF [9] uses small randomly initialized neural networks to project the bootstrap sample into a new feature space and builds an IF on each of these transformed representations. The resulting splits correspond to nonlinear random splitting functions in the original features.

[1]To ensure reproducibility, we provide full access to the code, including experiment scripts, hyperparameters and datasets used, at: https://github.com/AMCO-UniPD/KNNIFE

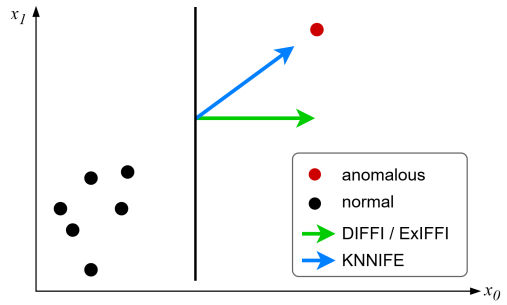

Fig. 1: Illustrating example on a two-dimensional dataset with a single anomaly (red point). The isolation tree split corresponds to the vertical line. DIFFI and ExIFFI impute feature importance using the split normal vector (green arrow) while KNNIFE distributes the importance along the data-dependent direction (blue arrow).

B. DIFFI, ExIFFI and FUBIFFI

DIFFI [10] is a post-hoc feature importance method specific for standard IF. Its global version quantifies the contribution of each feature based on how effectively it separates predicted outliers from inliers within the trees. At each internal node, an imbalance coefficient is computed, measuring how unevenly the samples of each class are divided between the child nodes. Features that frequently appear in nodes with high imbalance for outliers and low imbalance for inliers are assigned higher importance. In addition, DIFFI accounts for the depth at which samples are isolated: splits that isolate outliers closer to the root while inliers are deeper in the tree are considered more influential. For local feature importance, DIFFI does not consider node imbalance but only the depth at which each instance is isolated in each tree. Features appearing in shorter paths across the trees are assigned higher importance, as they contribute to isolating the instance earlier in the forest. We refer readers to [10] for a detailed explanation.

ExIFFI [11] extends DIFFI to explain EIF, handling splits that are not axis-aligned, such as oblique hyperplanes. The major modification is that the imbalance of each split can no longer be attributed to a single feature, as in DIFFI. Instead, the imbalance is distributed across all features involved in the split, proportionally to their contribution to the hyperplane. In particular, ExIFFI uses the normalized components of the hyperplane normal vector as weights to distribute the imbalance. Moreover, ExIFFI drops the explicit use of sample depth, since the imbalance along the path already captures the effect of early isolation, making depth information redundant. Local feature importance is computed by aggregating the weighted imbalance along the paths followed by the instance in the forest. Since KNNIFE follows the same procedure as ExIFFI, replacing only the hyperplane normal vector with a data-induced direction, we omit the algorithm of ExIFFI here and refer readers to Section III.

Finally, FUBIFFI [12] further extends this approach to nonlinear splits by defining the importance vector as the

mean of the gradient of the (differentiable) split function, computed in the data points of the split. The gradient serves as a proxy for the direction toward the decision boundary, which collapses to the normal vector of the splitting hyperplane in the IF and EIF cases. Although this approach generalizes to any differentiable splitting function, computing the gradient becomes increasingly expensive as the dimensionality, the number of data points and the complexity of the splitting function grow. However, none of these approaches account for the local data distribution at a node, and thus cannot adapt the importance direction to the data structure, as illustrated in Fig. 1.

III. METHODOLOGY

We argue that model-centric explanations produced by DIFFI, ExIFFI, and FUBIFFI do not necessarily identify the features most responsible for a point anomalousness. These methods explain which features the model used to isolate instances, but this does not necessarily correspond to the features that make the instance anomalous with respect to the data distribution.

To build intuition, consider the two-dimensional toy example illustrated in Fig. 1. Suppose we train an IF consisting of a single tree with max depth equal to 1. Assume the root split (the black vertical line) happens to isolate the anomalous instance and that the split is performed along x_0. In this case, DIFFI and ExIFFI assign maximal importance to x_0 and zero importance to x_1. This attribution is fully consistent with the model: the isolation of the point in that tree depends only on x_0 since there is no explicit involvement of x_1. However, this explanation fails to recognize that the anomalousness can also be attributed to x_1. If the number of trees increases, additional splits, potentially along x_1, are likely to appear across the IF. In some trees, the anomaly may be isolated through splits on x_1, leading DIFFI and ExIFFI to assign nonzero importance to both features. Nevertheless, for a finite number of trees and especially in high-dimensional settings, relevant features may be under-sampled. In such cases, DIFFI and ExIFFI importance scores depend on the random feature selection mechanism rather than on the actual factors that make the point anomalous in the data. We therefore propose a method that explicitly incorporates distributional properties of the dataset alongside the model structure, producing feature attributions that better reflect the data and can better support downstream root cause analysis.

A. KNNIFE Algorithm

KNNIFE follows the same algorithmic template as ExIFFI, replacing the definition of the split-level importance direction vector. Instead of the hyperplane normal, KNNIFE computes a *data-induced vector* $\hat{\mathbf{u}}(v)$ at each internal node v that encodes the feature-wise differences between nearest cross-partition neighbors.

Data-Induced Vectors: Let v be an internal node with associated data $\mathcal{D}_v$, and let $\mathcal{D}_{v_l}$ and $\mathcal{D}_{v_r}$ denote the data assigned to its left and right children, respectively. For each point $\mathbf{x} \in \mathcal{D}_{v_l}$, define its nearest neighbor in the opposite partition as $\mathrm{NN}(\mathbf{x}|\mathcal{D}_{v_r}) = \arg\min_{\mathbf{y}\in\mathcal{D}_{v_r}} \|\mathbf{x}-\mathbf{y}\|$, and analogously for points in $\mathcal{D}_{v_r}$. The (unnormalized) data-induced vector is:

$$\mathbf{u}(v) = \sum_{\mathbf{x}\in\mathcal{D}_{v_l}} \frac{|\mathbf{x}-\mathrm{NN}(\mathbf{x}|\mathcal{D}_{v_r})|}{\|\mathbf{x}-\mathrm{NN}(\mathbf{x}|\mathcal{D}_{v_r})\|} + \sum_{\mathbf{y}\in\mathcal{D}_{v_r}} \frac{|\mathbf{y}-\mathrm{NN}(\mathbf{y}|\mathcal{D}_{v_l})|}{\|\mathbf{y}-\mathrm{NN}(\mathbf{y}|\mathcal{D}_{v_l})\|}, \tag{4}$$

where $|\cdot|$ denotes element-wise absolute value. The normalized data-induced vector is $\hat{\mathbf{u}}(v) = \mathbf{u}(v)/\|\mathbf{u}(v)\|$.

Each term in Eq. (4) is a unit vector pointing in the direction (in absolute value) that most differentiates a point from its closest counterpart across the partition boundary. By aggregating these vectors over all points, $\hat{\mathbf{u}}(v)$ yields a vector whose components reflect the relative contribution of each feature in the separation induced by the split at node v (e.g., the blue arrow in Fig. 1). Note that ExIFFI derives importance directly from the hyperplane normal and is therefore independent of the data distribution associated with the internal node. KNNIFE instead uses the split only to induce a binary partition and derives the importance vector from the data distribution within that partition. As a consequence, it can be defined regardless of the split type, whether axis-aligned, hyperplane or nonlinear, allowing KNNIFE to be applied to any variant of IF.

Local Feature Importance (LFI): Local KNNIFE computes feature importance scores that summarize each feature contribution to anomaly detection for a specific data point. The procedure is as follows.

1) **Compute data-induced vectors.** For each internal node v in each tree t of each forest $\mathcal{F}^{(k)}, k \in [r]$, where r is the number of forests in the model (one for IF/EIF, possibly multiple for DIF, r), compute the normalized data-induced vector $\hat{\mathbf{u}}(v)$ via Eq. (4).
2) **Compute imbalance coefficients.** For each internal node v, define the imbalance coefficients for points routed to the left and right children as:

$$\lambda_{v_l} = \frac{n(v)}{n(v_l)}\hat{\mathbf{u}}(v), \qquad \lambda_{v_r} = \frac{n(v)}{n(v_r)}\hat{\mathbf{u}}(v), \tag{5}$$

where $n(v)$, $n(v_l)$, and $n(v_r)$ denote the number of points at node v and its children. Features receive higher importance when the split is imbalanced (i.e., one child receives far fewer points) and when the data-induced vector assigns high weight to those features.
3) **Compute LFI.** Compute the importance vector and normalization factor as

$$\mathbf{I}^{\mathbf{x}} = \sum_{k=1}^{r} \sum_{t\in\mathcal{F}^{(k)}} \sum_{v\in\mathrm{path}(\mathbf{x},t)} \lambda_{\mathrm{child}(\mathbf{x},v)} \tag{6}$$

$$\mathbf{V}^{\mathbf{x}} = \sum_{k=1}^{r} \sum_{t\in\mathcal{F}^{(k)}} \sum_{v\in\mathrm{path}(\mathbf{x},t)} \hat{\mathbf{u}}(v), \tag{7}$$

where $\mathrm{child}(\mathbf{x}, v)$ denotes the child of v containing $\mathbf{x}$. Finally, compute the local feature importance as:

$$\mathrm{LFI}(\mathbf{x}) = \mathbf{I}^{\mathbf{x}} / \mathbf{V}^{\mathbf{x}}. \tag{8}$$

The normalization factor $\mathbf{V}^{\mathbf{x}}$ accounts for the varying frequency with which different features appear in the data-induced vectors. The LFI vector identifies which features are most responsible for the anomalous classification of a specific instance, directly actionable information for industrial operators investigating individual alarms.

Global Feature Importance (GFI): Global KNNIFE computes feature importance scores that summarize each feature contribution to anomaly detection across the entire dataset. The procedure is analogous to the computation for LFI.

1) **Partition predictions.** For each forest $\mathcal{F}^{(k)}$, compute the sets of predicted inliers $\mathcal{P}_I^{(k)}$ and predicted outliers $\mathcal{P}_O^{(k)}$, for $k = 1, 2, \ldots, r$.
2) **Accumulate importance.** Compute the global importance vectors for inliers and outliers separately, as:

$$\mathbf{I}_I^{(k)} = \sum_{\mathbf{x}\in\mathcal{P}_I^{(k)}} \sum_{t\in\mathcal{F}^{(k)}} \sum_{v\in\text{path}(\mathbf{x},t)} \lambda_{\text{child}(\mathbf{x},v)} \tag{9}$$

$$\mathbf{I}_O^{(k)} = \sum_{\mathbf{x}\in\mathcal{P}_O^{(k)}} \sum_{t\in\mathcal{F}^{(k)}} \sum_{v\in\text{path}(\mathbf{x},t)} \lambda_{\text{child}(\mathbf{x},v)}. \tag{10}$$

3) **Compute GFI.** The global feature importance for each forest is: $\text{GFI}^{(k)} = (\mathbf{I}_O^{(k)}\mathbf{V}_I^{(k)})/\ (\mathbf{V}_O^{(k)}\mathbf{I}_I^{(k)})$ where divisions and products are element-wise. The overall model GFI is obtained by summation:

$$\text{GFI} = \sum_{k=1}^{r} \text{GFI}^{(k)}. \tag{11}$$

Features that strongly isolate outliers while minimally affecting inliers receive the highest GFI scores.

B. Computational Complexity

The dominant cost of KNNIFE lies in the one-time-only computation of the data-induced vectors, which involves three operations for each internal node v. First, the pairwise distances between the node samples is computed, with overall computational cost $O(\psi^2)$ per tree. Second, the cross-partition nearest neighbor for each node sample is retrieved from the precomputed distances. Having that the children of v are associated with $|\mathcal{D}_{v_l}|$ and $|\mathcal{D}_{v_r}|$ points, KNNIFE finds the cross-partition nearest neighbors using the precomputed distances with $|\mathcal{D}_{v_l}| \times |\mathcal{D}_{v_r}|$ lookups. Considering all nodes in a tree, this amounts to a complexity of $O(\psi^2)$ per tree. Third, the absolute difference vectors are accumulated into $\mathbf{u}(v)$, with $|\mathcal{D}_v|$ vector operations per node, resulting to $O(\psi)$ operations per tree. Notably, all of these quantities are computed just once, when fitting the explainer. Then, KNNIFE computes the local feature importance with virtually no overhead compared to the inference stage of isolation-based models.

IV. Experiments

We evaluate KNNIFE on 19 real-world anomaly detection benchmarks from Outlier Detection DataSets (ODDS)[13], on the industrial simulated dataset Tennessee-Eastman-Process (TEP), a standard benchmark for chemical-process fault detection, focusing on fault IDs 6 and 12, and on causRCA [14], a real-world dataset for root cause analysis recorded from the CNC control of an industrial vertical lathe. This specific datasets were chosen because they are among the few with ground-truth annotations for root cause, allowing proper evaluation of feature importance scores.

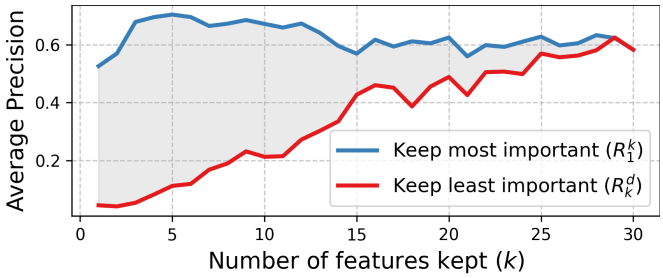

Fig. 2: For each number k of selected features, $\text{AP}(\mathcal{R}_1^k)$ and $\text{AP}(\mathcal{R}_k^d)$ are plotted. AUC_{FS} is the sum of their point-wise difference. Example using KNNIFE on wbc (IF).

We assess KNNIFE for explaining IF, EIF and DIF, comparing it with DIFFI when explaining IF and ExIFFI when explaining EIF. FUBIFFI is not included due to its high computational cost and reported poor performance on DIF [12], while it reduces exactly to ExIFFI when applied to EIF and IF. Results are aggregated over multiple random seeds. However, explanations produced by all methods are computed using the same anomaly detection model instance (same seed) to ensure fair comparison.

A. Evaluation metrics

We assess the anomaly detection performance of the underlying isolation-based model using Average Precision (AP) (area under the precision-recall curve), which is well-suited for highly imbalanced datasets and does not depend on a specific threshold for the anomaly score. This evaluation is crucial for DIFFI and ExIFFI, because when comparing their feature rankings with the ground-truth most important features, a high-performing base model ensures that the explanations are meaningful and reflect the dataset true structure.

For datasets from ODDS, where ground-truth feature importance is unavailable, correctness of the importance scores is evaluated using the standard proxy task of *feature selection*, as in previous work [10], [11], [12]. In this approach,the assigned importance scores induce a ranking $\mathcal{R} = [f_1, f_2, \ldots, f_d]$, where f_1 is the most important feature and f_d the least important. We denote by $\mathcal{R}_i^j = \{f_i, f_{i+1}, \ldots, f_j\}$ the subset of features from rank i to rank j in this ordering, and by $\text{AP}(\mathcal{R}_i^j)$ the AP of a model trained using only those features. The quality of the ranking is assessed by comparing two feature selection strategies:

- **Most important first**: progressively include top-ranked features, computing $\text{AP}(\mathcal{R}_1^k)$ for $k = 1, \ldots, d$.
- **Least important first**: progressively include bottom-ranked features, computing $\text{AP}(\mathcal{R}_k^d)$ for $k = d, \ldots, 1$.

These two curves are summarized by the **Area Under the**

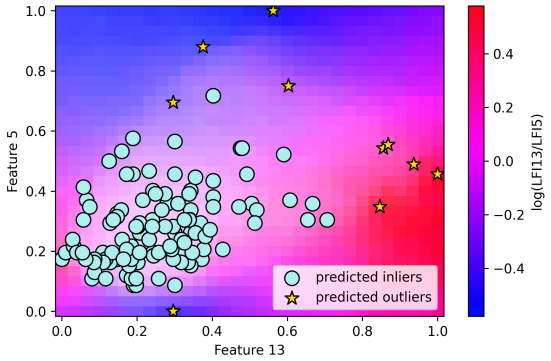

Fig. 3: Map of the local feature importance scores for EIF explained by KNNIFE on Wine dataset. The saturation represents the magnitude of the maximum importance. Hue represents the log-ratio importance of Feature 13 over Feature 5 (red where 13 dominates, blue where 5 dominates).

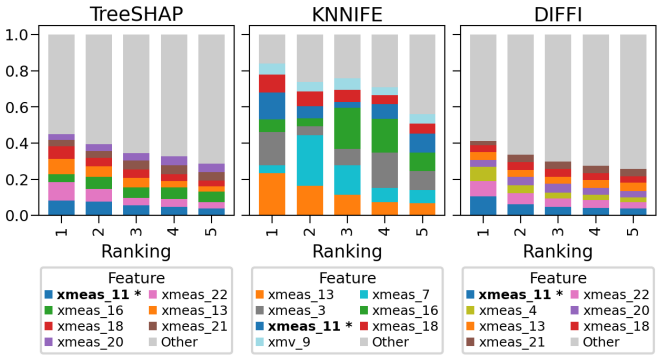

Fig. 4: Local feature rankings on anomalies for TreeShap, KNNIFE and DIFFI using IF on TEP fault 12. The heights of each sub-bar represents the percentage of predicted anomalies in which the corresponding feature appears at given rank position. Bold in the legend indicates the root-cause feature.

Curve of Feature Selection (AUC_{FS}) [11], defined as:

$$\mathrm{AUC}_{FS} = \sum_{k=1}^{d-1} \mathrm{AP}(\mathcal{R}_1^{d-k}) - \sum_{k=1}^{d} \mathrm{AP}(\mathcal{R}_k^d). \qquad (12)$$

A higher AUC_{FS} indicates that the importance scores effectively distinguish informative features from irrelevant ones: retaining the top-ranked features preserves detection performance as long as possible, while retaining only the bottom-ranked ones should degrade it the fastest. A visual explanation is provided in Fig. 2.

For TEP and causRCA, where the root-cause features of the fault are explicitly known, we report their position in the ranking. Ideally, these features should appear in top positions, indicating that the explanation method correctly identifies them.

B. Results

Local feature importance: KNNIFE local feature importance scores indicate most relevant feature for a specific data point. Fig. 3 shows, on the Wine dataset with EIF, a heatmap of the local scores of the two globally most important features (13 and 5) over the subspace spanned by these features. Predicted inliers cluster in a low-importance region (less saturated area) while outliers lie in saturated areas whose dominant color varies with spatial location. It shows that KNNIFE adapts explanations to the position of each anomaly position and they align with intuition (e.g., in area where only feature 13 deviates from normal, its importance dominates).

For TEP fault ID 12, Fig. 4 shows the rankings of the features for predicted anomalies based on local importance scores produced by KNNIFE, DIFFI and SHAP [15], a widely used model-agnostic feature attribution method, on IF. The true root cause, `xmeas_11`, corresponding to the product separator temperature, is ranked by all methods in the top positions in a considerable percentage of anomalies (blue sub-bars), although KNNIFE identifies it in a larger portion of anomalies compared to the other two. Moreover, KNNIFE frequently ranks `xmeas_13` highly in many samples. This feature is the product separator pressure, which is tightly coupled with the temperature. As a result `xmeas_13` is directly influenced by `xmeas_11` in the process dynamics [16], making it plausible for both variables to be truly relevant for the fault.

Global feature importance: Tab. I shows AP and AUC_{FS} for the ODDS datasets. KNNIFE achieves higher average AUC_{FS}, surpassing DIFFI (on IF) and ExIFFI (on EIF) on 12 out of 19 datasets. Moreover, we report results obtained by applying KNNIFE to DIF to demonstrate its applicability to a different variant of IF (and, more generally, to other isolation-forest-based models). We hypothesize that this improvement is due to a better alignment between KNNIFE and the feature-selection evaluation task. Since the proxy task trains a new model on the selected feature subset, it rewards methods that identify features genuinely relevant to anomaly separation in the data, rather than features that merely played a role in the original model internal splitting decisions. Negative metric values may be related to different phenomena. If truly important features exist, this may suggest that the feature ranking is of poor quality. When important features do not exist, negative metric values that are close to zero may reflect an arbitrary ordering of the features.

Tab. II reports the positions of the *true root-cause* features in the global rankings produced by KNNIFE, DIFFI and ExIFFI for TEP and causRCA. Notably, KNNIFE frequently ranks the true root-cause feature at the top and consistently places them higher than ExIFFI, supporting its potential utility in root cause analysis. While the difference with DIFFI is less pronounced, with cases where DIFFI surpasses KNNIFE (TEP 12 and cRCA 1, cRCA 2, cRCA 5, cRCA9 for causRCA in Tab. II), DIFFI is limited to standard IF and cannot be applied to other isolation-based anomaly detectors. We observed that when the AP is low (e.g., TEP 12 with EIF and DIF), the quality of explanations also suffers, since all explainers rely on the anomaly detector ability to distinguish

TABLE I: Comparison of mean AP and AUC_{FS} scores for IF, EIF and DIF.

Dataset	IF			EIF			DIF	
	AP	$AUC_{FS}\uparrow$		AP	$AUC_{FS}\uparrow$		AP	$AUC_{FS}\uparrow$
		DIFFI	KNNIFE		ExIFFI	KNNIFE		KNNIFE
wine	0.21	0.26	**3.08**	0.85	-8.92	**5.03**	0.85	5.11
glass	0.61	1.22	**1.59**	0.64	-0.51	**-0.06**	0.65	-0.13
vertebral	0.09	0.00	**0.03**	0.10	0.02	**0.05**	0.10	0.02
ionosphere	0.81	3.28	**4.29**	0.82	**-1.37**	-3.00	0.81	-2.68
wbc	0.60	-2.92	**7.89**	0.49	-2.37	**0.05**	0.53	0.64
breastw	0.97	**0.06**	-0.22	0.95	-0.20	**-0.03**	0.95	-0.01
pima	0.50	-0.44	**-0.14**	0.50	0.13	**0.13**	0.50	0.18
shuttle	0.05	**1.06**	-1.99	0.03	-1.77	**0.02**	0.03	0.12
vowels	0.16	**0.55**	-0.19	0.15	**0.58**	-0.51	0.12	-0.76
letter	0.09	**0.12**	-0.67	0.09	**0.15**	-0.06	0.09	0.12
cardio	0.57	**4.13**	3.28	0.56	**3.39**	3.16	0.55	3.10
thyroid	0.57	2.40	**3.00**	0.23	1.78	**2.13**	0.23	2.16
optdigits	0.05	-1.70	**-0.02**	0.04	-1.38	**0.11**	0.04	-0.22
pageblocks	0.47	**1.90**	-1.35	0.47	**-0.51**	-2.32	0.46	-1.77
satimage-2	0.92	3.74	**13.16**	0.95	3.42	**13.78**	0.94	14.15
satellite	0.67	4.11	**4.28**	0.68	3.03	**4.60**	0.68	4.72
pendigits	0.29	0.83	**3.01**	0.28	**1.48**	0.28	0.19	-0.07
annthyroid	0.31	1.83	**2.06**	0.16	**1.36**	0.99	0.16	0.98
mammography	0.21	**0.79**	0.51	0.18	0.07	**1.02**	0.18	1.06
Mean		1.12	**2.19**		-0.09	**1.33**		1.41

TABLE II: True root-cause feature rank positions for IF, EIF, and KNNIFE (cRCA: causRCA fault IDs).

Dataset	IF			EIF			DIF	
	AP	Rank-pos↓		AP	Rank-pos↓		AP	Rank-pos↓
		DIFFI	KNNIFE		ExIFFI	KNNIFE		KNNIFE
TEP 6	0.96	17	**12**	0.94	33	**10**	0.94	10
TEP 12	0.91	**2**	5	0.68	23	**10**	0.67	11
cRCA 1	0.87	**59**	70	0.78	11	**1**	0.77	65
cRCA 2	0.90	**59**	72	0.88	10	**1**	0.88	63
cRCA 3	0.98	54,58	**51,51**	0.96	10,11	**1,1**	0.94	63,63
cRCA 4	0.65	63,**1**,49, 47,58	**13**,79,**13**, **13**,**13**	0.74	11,61,71, 67,68	**2**,**41**,**47**, **46**,**47**	0.73	3,55,3, 3,3
cRCA 5	1.00	**1**	39	1.00	10	**1**	1.00	68
cRCA 6	1.00	**1**,**1**	**1**,**1**	1.00	11,12	**1**,**1**	1.00	1,1
cRCA 7	1.00	**1**	**1**	1.00	10	**1**	1.00	1
cRCA 8	1.00	**1**,**1**	**1**,**1**	1.00	10,12	**1**,**1**	1.00	1,1
cRCA 9	0.87	**1**,**1**,**1**	56,**1**,58	0.79	60,8,11	**31**,**1**,**1**	0.79	65,1,69
cRCA 10	0.90	**1**,**1**,**1**	**1**,**1**,**1**	0.86	6,10,13	**1**,**1**,**1**	0.87	1,1,1

anomalies.

Fig. 5 compares the normalized global feature importance distributions produced by ExIFFI and KNNIFE on TEP 12 with EIF. KNNIFE yields a more selective importance profile, concentrating high scores on a small subset of features while assigning lower importance to the rest. In contrast, ExIFFI produces a flatter curve with high variance across runs: this makes it difficult to distinguish truly important features from irrelevant ones, meaning that for a low number of trees the resulting ranking can change substantially between runs. Because KNNIFE is both more stable and more discriminative, it provides a more reliable and interpretable feature ranking. This property is especially valuable in industrial scenarios like TEP, where practitioners require consistent and well-separated importance scores to confidently identify root causes and act upon them.

V. CONCLUSIONS

We introduce KNNIFE, a novel feature-importance method for any isolation-based anomaly detection model

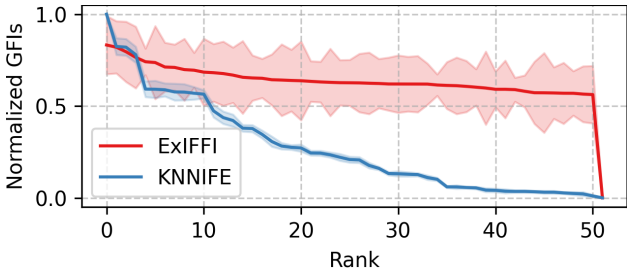

Fig. 5: Normalized GFI distributions on TEP 12 (EIF). Both importance vectors are normalized to unit norm. KNNIFE produces a sharper, more selective importance curve with substantially lower inter-run variance compared to ExIFFI.

that leverages dataset properties rather than relying only on the model internal structure. Experiments on real-world, simulated, and industrial datasets demonstrate that KNNIFE produces feature importance scores better aligned with true root causes. A limitation is that KNN can be affected by the curse of dimensionality, which may impact performance in high-dimensional datasets and will be addressed in future work.

REFERENCES

[1] V. Chandola, A. Banerjee *et al.*, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, Jul. 2009.

[2] L. C. Brito, G. A. Susto *et al.*, "An explainable artificial intelligence approach for unsupervised fault detection and diagnosis in rotating machinery," *Mech. Syst. Signal Process.*, vol. 163, p. 108105, 2022.

[3] A. Almalawi, X. Yu *et al.*, "An unsupervised anomaly-based detection approach for integrity attacks on scada systems," *Comput. Secur.*, vol. 46, pp. 94–110, 2014.

[4] F. T. Liu, K. M. Ting *et al.*, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.

[5] Y. Cao, H. Xiang *et al.*, "Anomaly detection based on isolation mechanisms: A survey," *Mach. Intell. Res.*, vol. 22, no. 5, pp. 849–865, 2025.

[6] R. Bouman, Z. Bukhsh *et al.*, "Unsupervised anomaly detection algorithms on real-world data: how many do we need?" *J. Mach. Learn. Res.*, vol. 25, no. 105, pp. 1–34, 2024.

[7] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "Adbench: Anomaly detection benchmark," *Advances in neural information processing systems*, vol. 35, pp. 32 142–32 159, 2022.

[8] S. Hariri, M. C. Kind *et al.*, "Extended isolation forest," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1479–1489, 2021.

[9] H. Xu, G. Pang *et al.*, "Deep isolation forest for anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12 591–12 604, 2023.

[10] M. Carletti, M. Terzi *et al.*, "Interpretable anomaly detection with diffi: Depth-based feature importance of isolation forest," *Eng. Appl. Artif. Intell.*, vol. 119, p. 105730, 2023.

[11] A. Arcudi, D. Frizzo *et al.*, "Enhancing interpretability and generalizability in extended isolation forests," *Eng. Appl. Artif. Intell.*, vol. 138, p. 109409, 2024.

[12] A. Arcudi, A. Ferreri *et al.*, "Function based isolation forest (fubif): A unifying framework for interpretable isolation-based anomaly detection," *IFAC-PapersOnLine*, vol. 59, no. 26, pp. 91–96, 2025.

[13] S. Rayana, "ODDS library," 2016.

[14] C. W. Mehling, S. Piepere *et al.*, "causRCA: Real-World Dataset for Causal Discovery and Root Cause Analysis in Machinery," July 2025.

[15] S. M. Lundberg, G. Erion *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nat. Mach. Intell.*, vol. 2, no. 1, pp. 56–67, 2020.

[16] M. G. Don and F. Khan, "Dynamic process fault detection and diagnosis based on a combined approach of hidden markov and bayesian network model," *Chem. Eng. Sci.*, vol. 201, pp. 82–96, 2019.