Unsupervised Outlier Detection: New Approaches for Robust Ensembles and Interpretability

Vasileios Papastefanopoulos

A dissertation submitted for the degree of

Doctor of Philosophy

Department of Mathematics

University of Patras

May 6, 2025

I, Vasileios Papastefanopoulos, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Outlier detection (OD) is crucial for identifying anomalies in fields like fraud detection, network security, and medical diagnostics. However, model selection and interpretability remain significant challenges in unsupervised settings. This thesis addresses these challenges by introducing methodologies for improved model selection and interpretability. First, a meta-learning ensemble framework is introduced for improved model selection. This framework treats outlier scores from multiple detectors as features and employs unsupervised feature selection to identify the most informative detectors for a given dataset. The selected detectors are then combined through an ensemble voting approach. Second, a taxonomy of interpretability methods is presented. This taxonomy categorizes existing techniques based on their scope (local vs. global), model dependency (model-agnostic vs. model-specific), and whether they are applied during or after model training (intrinsic vs. post hoc). The taxonomy also considers applicability across diverse data types, including tabular, text, image, and graph data. Finally, an unsupervised OD method is proposed that leverages both the density estimation capabilities of normalizing flows and the interpretability of decision trees. The method uses normalizing flows to generate pseudo-labels, which then train a decision tree classifier. This provides clear, rule-based classification boundaries for outlier detection, informed by the underlying data density. These contributions collectively facilitate the development of unsupervised OD systems with improved model selection, enhanced interpretability, and a clearer understanding of the underlying decision-making process, ultimately supporting the deployment of these systems in critical real-world applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Outlier detection, or anomaly detection, involves identifying data points that significantly deviate from expected patterns in a dataset. These anomalies, often driven by different underlying processes, typically comprise less than 5% of the data [1]. Although there is no universally accepted definition of outliers [2], their detection is fundamental to data analysis, enabling identification and, when necessary, removal of data points that could distort meaningful patterns.

Outlier detection is critical for numerous applications where anomalies signal significant or harmful events. In finance, it detects fraudulent transactions [3, 4], while in healthcare, it helps diagnose conditions such as cancer through medical imaging [5, 6]. Similarly, telecommunications use it to uncover fraud, and industrial systems rely on it to predict equipment failures, minimizing costly downtime [2]. Additional applications include IoT systems [7], structural health monitoring [8], and network intrusion detection, where challenges often involve processing large, high-dimensional data streams [9, 10].

From a data science perspective, managing outliers ensures the integrity of machine learning and statistical models, allowing algorithms to capture trends and insights more accurately [11, 12]. Although outliers can distort models, they also serve as valuable signals, revealing unexpected or novel system behaviors [13]. This dual role complicates outlier detection: Some anomalies need to be filtered to preserve data quality, while others require a deeper analysis to uncover their causes. In safety-critical domains such as fraud detection or system malfunction monitoring,

distinguishing between these types is essential [14, 15].

Despite its critical importance, outlier detection remains an open challenge, especially with high-dimensional, noisy, and evolving datasets. Robust methodologies are essential for accurately identifying and understanding anomalies, enabling actionable insights, and enhancing model performance.

1.1 Outlier Detection Methods

To address the varied nature of outliers, machine learning frameworks for outlier detection are broadly classified into supervised, semi-supervised, and unsupervised methods, each with distinct advantages and limitations [2, 16]. The choice of method often depends on the availability of labeled data and the dataset's characteristics.

1.1.1 Supervised Outlier Detection

Supervised outlier detection trains classifiers on labeled data to learn patterns that distinguish outliers from normal points. These methods, such as support vector machines (SVMs) and neural networks, often achieve high accuracy by generalizing from the labeled examples [17]. For example, they are widely used in domains like fraud detection.

However, their reliance on labeled datasets limits their applicability in real-world scenarios, where anomalies are rare and labeling requires domain expertise. Furthermore, labeled data may not encompass all types of potential anomalies, leading to models that struggle with novel or unexpected outliers [18]. This dependence on comprehensive and representative labeled data poses significant challenges in dynamic and complex environments.

1.1.2 Semi-Supervised Outlier Detection

Semi-supervised methods address the limitations of supervised approaches by using a small set of labeled examples to define a "normality boundary," which is then refined with unlabeled data [19]. This allows semi-supervised methods, such as one-class SVMs, to leverage abundant unlabeled data while maintaining some level of guidance from labeled examples.

However, the success of these methods depends heavily on the quality and representativeness of the labeled examples. Poorly curated labels can result in boundaries that do not detect novel anomalies or exclude valid patterns. Semi-supervised methods are most effective when labeled data, though limited, is carefully curated to capture the diversity of normal behaviors.

1.1.3 Unsupervised Outlier Detection

Unsupervised outlier detection is the most practical approach in scenarios where labeled data is unavailable. These methods identify anomalies by analyzing the intrinsic structure of the dataset, independent of any labeling. Common techniques include:

- **Clustering-Based Methods:** Algorithms such as k-means and DBSCAN detect outliers as points that deviate from cluster centroids or reside in low-density regions [20].
- **Density-Based Methods:** Techniques like the Local Outlier Factor (LOF) estimate local densities and flag points in significantly sparser regions as anomalies.
- **Reconstruction-Based Methods:** Methods such as Principal Component Analysis (PCA) and autoencoders reconstruct data patterns, identifying points with high reconstruction errors as outliers.

Although unsupervised methods excel at handling heterogeneous high-dimensional datasets, they face challenges such as the curse of dimensionality, which complicates distance and density calculations [21]. Furthermore, without labeled data, distinguishing true anomalies from noise can be uncertain.

Despite these challenges, unsupervised methods are indispensable for fields such as network intrusion detection, fraud prevention, and medical diagnostics, where labeled data are scarce or unreliable. Their flexibility and broad applicability make them the foundation for modern anomaly detection frameworks.

Each of these frameworks offers valuable tools for outlier detection, but all face inherent challenges, particularly in the absence of labeled data. Addressing issues

such as model selection, interpretability, and scalability is critical to advancing the field. These challenges form the basis for the methodologies and solutions explored in this thesis.

1.2 Challenges and Motivation

Outlier detection is a critical task in data analysis, particularly in unsupervised settings where labeled data is unavailable. Despite the development of numerous unsupervised methods, several challenges hinder their practical application and motivate further research.

The first challenge is model selection. Without ground-truth labels, traditional evaluation and hyperparameter optimization methods are inapplicable. Different algorithms rely on varying assumptions and measures of "outlierness," such as density [22–24], distance [25, 26], or angle [27], leading to disparate predictions in the same dataset. The complexity of modern datasets, which are often heterogeneous and high-dimensional, further complicates this process, making it difficult to identify the most suitable algorithm for a given dataset.

The second challenge lies in the lack of interpretability in unsupervised outlier detection. As the demand for explainable AI (XAI) grows, understanding the reasoning behind detection decisions has become critical, particularly in high-stakes domains such as healthcare, finance, and security. Unlike supervised models, where ground-truth labels enable interpretability, unsupervised models remain opaque [28]. Most algorithms provide little insight into their predictions, locally (for individual points) or globally (across datasets), often relying on post hoc analyses or visualizations to explain the results [29]. This lack of transparency limits its trust, usability, and adoption in critical applications where accountability is paramount.

These challenges highlight the need for novel approaches that enhance the robustness of outlier detection, enable effective model selection without labeled data, and provide greater interpretability. Addressing these issues is a central aim of this thesis, which proposes new methods to make unsupervised outlier detection more

reliable, transparent, and deployable in critical real-world applications.

1.3 Thesis Contributions and Structure

This thesis introduces three main contributions to the field of unsupervised outlier detection, each aiming to overcome the limitations outlined in the previous section and each published in peer-reviewed journals.

1. Towards Robust Ensembles: A Meta-Learning Approach for Unsupervised Model Selection.

The first contribution is a meta-learning framework that focuses on model selection and robustness to noisy outlier scores in unsupervised outlier detection. In unsupervised settings, individual outlier detection algorithms often produce inconsistent or inaccurate scores due to varying assumptions and sensitivities to different data characteristics. These unreliable scores can be considered "noisy data," and they can significantly degrade the performance of ensemble methods that combine multiple algorithms. This framework addresses this challenge by employing unsupervised feature selection in the construction of ensembles, mitigating the weaknesses of individual algorithms while leveraging their collective strengths. It treats outlier scores from individual algorithms as features and analyzes them to identify the most informative models for a given dataset. This process serves as a model selection mechanism. Unlike traditional ensemble methods that often combine models arbitrarily, the proposed framework prioritizes models based on their relevance to the specific dataset, as determined through unsupervised feature selection. This is crucial in unsupervised settings where ground-truth labels for model evaluation are unavailable. The selected algorithms are then weighted and combined within the meta-learning framework, which adapts to the dataset's underlying structure without relying on ground-truth labels. The ensemble voting approach further enhances the robustness and accuracy of the framework. The details of this methodology, including the specific feature selection techniques and weighting mechanisms, are elaborated in Chapter 2. The chapter also includes experimental results on diverse datasets, including heterogeneous and high-dimensional scenarios, demonstrating

the framework's ability to improve outlier detection performance compared to individual algorithms and traditional ensemble methods. A version of this work has been published as "*Papastefanopoulos, V., Linardatos, P., & Kotsiantis, S. (2021). Unsupervised Outlier Detection: A Meta-Learning Algorithm Based on Feature Selection. Electronics, 10(18), 2236*" (ref. [30]).

2. A Taxonomy for Machine Learning Interpretability Methods.

The second contribution is a taxonomy of interpretability methods, addressing the need for transparency in machine learning. This taxonomy categorizes existing techniques based on key dimensions, such as their scope (local vs. global), model dependency (model-agnostic vs. model-specific), and the timing of their application (intrinsic vs. post-hoc). These dimensions are crucial for understanding the capabilities and limitations of different interpretability approaches. Furthermore, it analyzes the applicability of these methods to various data types (text, image, tabular, graph), a dimension often overlooked in prior work. This consideration of data type is particularly important, as the optimal interpretability method can vary significantly depending on the nature of the data being analyzed. The taxonomy aims to provide a structure for researchers and practitioners in selecting suitable techniques, especially in domains such as healthcare, finance, and security, where understanding the reasoning behind model decisions is paramount for ensuring fairness, accountability, and trust. Chapter 3 provides a discussion of the taxonomy, an analysis of existing methods, including their strengths and weaknesses, and presents the classification framework. The chapter also includes illustrative examples of how to apply the taxonomy to select appropriate interpretability methods for specific scenarios. A version of this work has been published as "*Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable AI: A Review of Machine Learning Interpretability Methods. Entropy, 23(1), 18*" (ref. [31]).

3. An Interpretable Method Combining Normalizing Flows with Decision Trees.

The third contribution is an unsupervised outlier detection method that incorporates interpretability, a crucial aspect often lacking in existing methods. This method

uniquely combines the strengths of normalizing flows, for density estimation, with decision trees, for generating interpretable decision rules. The normalizing flows are first used to model the data distribution and provide density estimations, which are then transformed into pseudo-labels that differentiate between inliers and outliers. These pseudo-labels act as a bridge between the density estimation of the normalizing flows and the decision-making process of the decision tree. These pseudo-labels are subsequently used to train a decision tree. This approach allows for both global interpretability, by examining the overall structure of the decision tree, and local interpretability, by analyzing the specific rules that lead to a particular prediction. Chapter 4 presents this method and its application, including experimental results demonstrating its ability to achieve high detection accuracy while maintaining interpretability, a balance that is often difficult to achieve in unsupervised settings. A version of this work has been published as "*Papastefanopoulos, V., Linardatos, P., & Kotsiantis, S. (2025). Combining normalizing flows with decision trees for interpretable unsupervised outlier detection. Engineering Applications of Artificial Intelligence, 141, 109770*" (ref. [32]).

These three main contributions of this thesis—the meta-learning framework, the interpretability taxonomy, and the novel OD method using normalizing flows and decision trees—provide researchers and practitioners with tools to develop more adaptable, transparent, and reliable outlier detection systems. By addressing the limitations of existing methods, this work contributes to the broader adoption of unsupervised OD in critical, real-world applications. Chapter 5 concludes the thesis by summarizing the key findings, discussing their implications, and suggesting directions for future exploration.

Chapter 2

Towards Robust Ensembles: A Meta-Learning Approach for Unsupervised Model Selection

Unsupervised outlier detection faces significant challenges, particularly the lack of labeled data, which complicates effective model selection and evaluation. Individual detection algorithms often struggle to generalize across diverse datasets due to their context-specific assumptions about data structure. This inconsistency highlights the need for frameworks that combine the strengths of multiple detection methods while addressing their limitations. To address this, this chapter introduces a feature selection-based meta-learning framework for robust ensembles. The framework systematically evaluates and prioritizes detection models through feature selection, filtering out less effective algorithms, and retaining only the most informative ones. This approach enhances both robustness and accuracy, reducing the influence of weaker detectors and adapting to diverse datasets, particularly in high-dimensional or noisy settings.

The key contributions of this chapter are as follows.

- **Feature Selection-Based Framework for Robust Ensembles**: Introduction of a novel framework that integrates multiple unsupervised outlier detection algorithms, providing a systematic approach to model selection and ensemble building.

- **Model Selection via Feature Selection**: Development of an unsupervised feature selection method, Fixed SPEC, to identify and retain the most effective detectors, reducing noise and improving ensemble performance.
- **Optimized Aggregation Strategy**: Implementation of the Average of Maximum (AOM) aggregation strategy to balance extreme outlier signals with model diversity for robust predictions.
- **Comprehensive Empirical Validation**: Validation on 17 datasets, demonstrating state-of-the-art performance in terms of AUC-ROC and precision, and surpassing 10 widely used outlier detection methods.

These contributions address critical challenges in unsupervised outlier detection by combining robust ensemble building with effective model selection. By focusing on adaptability, systematic model evaluation, and noise reduction, this framework lays the foundation for more reliable outlier detection in real-world scenarios.

The remainder of this chapter is organized as follows. Section 2.1 reviews related work on unsupervised outlier detection. Section 2.2 details the proposed framework, including the feature selection process. Section 2.3 describes the data and the experimental setup. Section 2.4 presents the benchmark results and discusses key findings. Finally, Section 2.5 concludes the chapter with a summary of the contributions and next steps.

2.1 Related Work

Ensemble methods have emerged as essential for enhancing unsupervised outlier detection, offering a way to increase accuracy and robustness, especially in the absence of labeled data. Aggarwal [33] highlighted the limitations of single-model detectors, advocating for ensembles to enhance accuracy and robustness. Based on this, Zimek et al. [34] identified key factors for the success of ensembles in unsupervised detection, while Mukhiya and Kumar [35] demonstrated the performance benefits of ensembles in high-dimensional datasets.

In recent years, meta-learning and internal evaluation metrics have provided

new avenues for refining ensemble performance. Marques et al. [36] explored internal metrics for effective model weighting in unsupervised settings, while Alexandropoulos et al. [37] and Zhao et al. [38] introduced meta-learning techniques for dynamic model optimization. Li et al. [39] applied ensemble clustering to non-traditional domains such as education, demonstrating the adaptability of these techniques.

For high-dimensional or noisy datasets, tailored ensemble techniques have shown promise. Chaurasia et al. [40] demonstrated that using a diverse ensemble of autoencoders significantly improved detection accuracy in high-dimensional data, illustrating the benefit of combining models with varying sensitivity to complex patterns. Ouyang et al. [41] addressed noise with EBOD, and Sarvari et al. [42] used boosting-based autoencoders for iterative improvements in outlier detection. Chakraborty et al. [43] refined ensemble performance by iteratively filtering noisy data. Cheng et al. [44] incorporated feature selection into ensembles, and Zhao et al. [45] tackled scalability with the SUOD framework for large datasets. Mukhriya and Kumar [46] improved detection by selectively tuning the ensemble parameters, particularly for datasets with varying densities. Merrill and Olson [47] applied ensemble kernel principal component analysis (PCA) for the detection of hyperspectral anomalies, showing that the combination of multiple kernel-PCA models improved the detection of complex, high-dimensional hyperspectral data.

Ensemble methods have also found success in domain-specific applications such as cybersecurity, where Boukela et al. [48] and Le and Zincir-Heywood [49] improved threat detection using ensembles. An et al. [50] combined autoencoders and GMMs for cyberattack detection, while Plakias and Boutalis [51] demonstrated the effectiveness of the ensemble for fault detection in industrial systems. Vosseler [52] applied ensembles to insurance fraud detection, while Khan and Haroon [53] used deep learning ensembles for the detection of social network anomalies.

Cluster-based ensembles have also been proven to be effective. Thomas and Judith [54] proposed a voting-based ensemble framework, while Ray et al. [55]

and Saha et al. [56] combined traditional clustering algorithms such as k-means and DBSCAN to capture various outlier patterns. Yu and Kang [57] introduced a novelty score based on the grouping ensemble, and Li et al. [58] improved outlier detection using empirical cumulative distribution functions (ECDF).

Innovative approaches have addressed feature selection and regional partitioning. Cheng et al. [59] used ensembles for social network analysis, while Yang et al. [60] introduced a regional ensemble framework that applies local detectors to different subsets of data, improving precision in heterogeneous datasets.

In time series data, ensembles have been effective in capturing dynamic anomalies. Campos et al. [61] developed a diversity-driven convolutional ensemble for time series detection, while Katser et al. [62] applied ensembles for changepoint detection, improving robustness against abrupt changes.

Lastly, adaptive and hybrid ensembles have emerged as powerful tools for evolving datasets. Siddiqui and Boukerche [63] introduced adaptive autoencoder ensembles for the detection of intrusion into IoT networks, dynamically adapting to new data. Kandanaarachchi [64] developed an item-response theory-based framework, dynamically adjusting model weights based on performance, demonstrating significant improvements in high-dimensional settings.

Despite the progress made, several limitations persist in ensemble-based unsupervised outlier detection. Most existing approaches either retain all models in the ensemble, regardless of individual performance, or lack mechanisms to optimally integrate the results of multiple algorithms in noisy, high-dimensional settings. Moreover, while advances in meta-learning and internal evaluation metrics have been explored, identifying the optimal selection and combination of outlier detection algorithms remains an unresolved issue, particularly due to the diverse and context-dependent performance of these algorithms across different datasets.

Our approach seeks to bridge these gaps by introducing a more selective and robust method to combine outlier detection algorithms. By evaluating the quality of outlier scores produced by each algorithm and selectively including only the most reliable models, our method enhances both accuracy and robustness across differ-

ent datasets. This selective approach provides a consistent detection framework in challenging unsupervised settings where labeled data are unavailable.

The following section outlines the proposed methodology and details how our approach refines the ensemble process to improve outlier detection.

2.2 Proposed Methodology

This chapter proposes a novel methodology for unsupervised outlier detection that combines spectral feature selection with ensemble aggregation techniques to enhance accuracy and robustness. The approach addresses the challenge of selecting reliable outlier detection models by retaining the most informative base models and integrating them into a robust ensemble framework.

The methodology begins by training multiple base outlier detectors, such as k-Nearest Neighbors (k-NN) [26], Local Outlier Factor (LOF) [22], and Cluster-Based Local Outlier Factor (CBLOF) [65] on standardized training data. Each detector generates confidence scores quantifying the outlierness of the data points. These scores are treated as features, and spectral feature selection is applied to identify and retain the most relevant detectors by analyzing the structure of the score vectors [66].

Spectral feature selection, grounded in spectral graph theory [67], analyzes the alignment of score vectors with the intrinsic structure of the dataset, retaining only the most informative detectors. This reduces noise and improves robustness, particularly in high-dimensional datasets [66].

The retained detector scores are then aggregated using the Average of Maximum (AOM) ensemble method [13]. AOM divides the selected detectors into subgroups, computes the maximum score within each group, and averages these maximum values to derive the final prediction. This strategy captures the strongest outlier signals while preserving diversity between models, ensuring robustness to noise and variations in data distribution.

By dynamically selecting the most relevant detectors and integrating spectral feature selection with AOM aggregation, the proposed methodology improves the

accuracy and robustness of unsupervised outlier detection. The key innovation lies in the adaptive selection of models, which optimizes ensemble performance in noisy and high-dimensional settings [13, 66].

Each stage of this process, including base model selection, feature selection, and score aggregation, contributes to a robust ensemble framework. Algorithm 1 presents the pseudocode for the methodology, and Figure 2.1 illustrates the workflow, providing a comprehensive overview of the proposed approach.

Algorithm 1 Overview of the ensemble-based outlier detection methodology, detailing steps from data preparation and base detector training to spectral feature selection, result aggregation, and performance evaluation.

Step 1: Input
Dataset $\mathscr{D}$
Number of detectors n
Predefined outlier percentage P_{outlier}
Step 2: Data Preparation
Standardize $\mathscr{D}$ to normalize input features.
Split $\mathscr{D}$ into training dataset $\mathscr{D}_{\text{train}}$ (60%) and testing dataset $\mathscr{D}_{\text{test}}$ (40%).
Step 3: Train Base Detectors
for $i = 1$ to n **do**
Train outlier detector OD_i on $\mathscr{D}_{\text{train}}$.
Generate outlier score S_i for all data points in $\mathscr{D}_{\text{train}}$ based on P_{outlier}.
end for
Step 4: Perform Spectral Feature Selection
Construct the outlier score matrix $S \in \mathbb{R}^{n \times m}$ from all detectors.
Compute spectral features using the Fixed SPEC algorithm.
Retain 60% of the most informative detectors based on spectral features.
Step 5: Aggregate Results Using Average of Maximum (AOM)
Divide retained detectors into subgroups.
Compute the maximum score for each subgroup.
Average these maximum scores to produce predictions for the training data.
Step 6: Apply the Trained Model to $\mathscr{D}_{\text{test}}$
Use the retained detectors to compute outlier scores for $\mathscr{D}_{\text{test}}$.
Apply AOM aggregation to generate final outlier scores for $\mathscr{D}_{\text{test}}$, based on P_{outlier}.
Step 7: Evaluate Performance
Compare predictions with ground truth in $\mathscr{D}_{\text{test}}$ to evaluate the model's performance.
Step 8: Output
Final aggregated outlier scores for $\mathscr{D}_{\text{test}}$
Predictions for $\mathscr{D}_{\text{test}}$, based on P_{outlier}
Evaluation metrics

The flow chart in Figure 2.1 visually outlines the methodology described in the

Figure 2.1: Flowchart of the proposed unsupervised outlier detection methodology, illustrating the steps from training base detectors to applying spectral feature selection and ensemble aggregation.

pseudocode. It highlights the step-by-step process, from training base detectors to applying spectral feature selection and ensemble aggregation.

With the methodology reviewed, the subsequent sections provide in-depth explanations of each step, from model selection to aggregation techniques that finalize the predictions of the ensemble.

2.2.1 Base Outlier Detection Models

The proposed method begins by training a diverse ensemble of unsupervised outlier detection algorithms, each designed to capture different aspects of anomaly patterns within the data. Specifically, the base models used in this ensemble include:

- **k-Nearest Neighbors (k-NN):** The k-NN algorithm [26] measures the dis-

tance between a data point and its k nearest neighbors to estimate the density around that point. The outlier score is calculated using an aggregation function, such as the maximum, mean, or median of the distances to the k nearest neighbors. Ideal for datasets where density variations can indicate anomalies.

- **Local Outlier Factor (LOF):** LOF [22] estimates the local density of a data point by comparing the density of the point to the densities of its neighbors. Points that have significantly lower local density than their neighbors are considered outliers. Effective in datasets with heterogeneous density regions.
- **Cluster-Based Local Outlier Factor (CBLOF):** CBLOF [65] first applies k-means clustering to split the data into clusters, then classifies clusters into "small" and "large." The outlier score is calculated based on the cluster size and the distance to the cluster centroid. For points in small clusters, the distance to the nearest large cluster is used instead. Well-suited for clustering-based anomaly identification.
- **Histogram-Based Outlier Score (HBOS):** HBOS [23] calculates outlier scores by assuming feature independence and modeling each feature's distribution with histograms. The histograms are then combined to estimate the overall outlier score for each data point. Efficient for high-dimensional datasets with independent features.
- **One-Class Support Vector Machines (OCSVM):** OCSVM [68] learns a decision boundary that separates normal data from outliers by training on the normal observations. The kernel function plays an important role, and the radial basis function (RBF) kernel is often used to capture complex data distributions. Useful for defining boundary-based anomalies with kernel functions.
- **Minimum Covariance Determinant (MCD):** MCD [69] estimates the covariance matrix of the dataset while minimizing the influence of outliers. It identifies outliers by computing the Mahalanobis distance from the estimated mean and covariance, with outliers being the points farthest from the center of the distribution. Focuses on data with normal distribution assumptions and robust distance-based scoring.

- **Principal Component Analysis (PCA):** PCA [70] is a dimensionality reduction technique that identifies directions of maximum variance in the data. The outlier score is derived from the reconstruction error, calculated as the sum of the squared distances between each point and its projection onto the principal components. Captures anomalies by evaluating variance directions.
- **Angle-Based Outlier Detection (ABOD):** ABOD [27] calculates the variance of angles between a data point and all other points. Points with lower variance in angles are likely to be outliers, as they lie far from the main clusters in the dataset. Works well for detecting angular anomalies in clusters.
- **Feature Bagging (FB):** Feature Bagging [71] is a meta-method that combines several base detectors (e.g., k-NN, LOF, ABOD) trained on random subsets of features. The predictions of these base detectors are averaged to produce a final outlier score, increasing robustness by reducing overfitting. Reduces overfitting by aggregating models trained on random feature subsets.
- **Isolation Forest (iForest):** Isolation Forest [72] isolates anomalies by recursively partitioning the feature space. Points that require fewer splits to be isolated are more likely to be outliers. Efficient for large datasets due to its recursive partitioning approach.

Each of these models generates an outlier score for every data point in the dataset. These scores reflect how anomalous each point is relative to the rest of the data, and by using multiple models, the ensemble can capture a wider range of anomaly types. By combining models with varying assumptions and mechanisms, the ensemble compensates for the limitations of individual models, resulting in a comprehensive approach to anomaly detection. However, the different assumptions and mechanisms of each model mean that not all perform equally well in every dataset. Therefore, in the next step, we apply unsupervised feature selection to filter out less effective models, ensuring that only the most relevant models contribute to the final prediction.

The decision boundaries in Figure 2.2 underscore the value of using models with different approaches to ensure comprehensive coverage of anomaly pat-

terns. This variation highlights the importance of selecting models that best suit the dataset's characteristics, which we achieve through the proposed feature selection method.

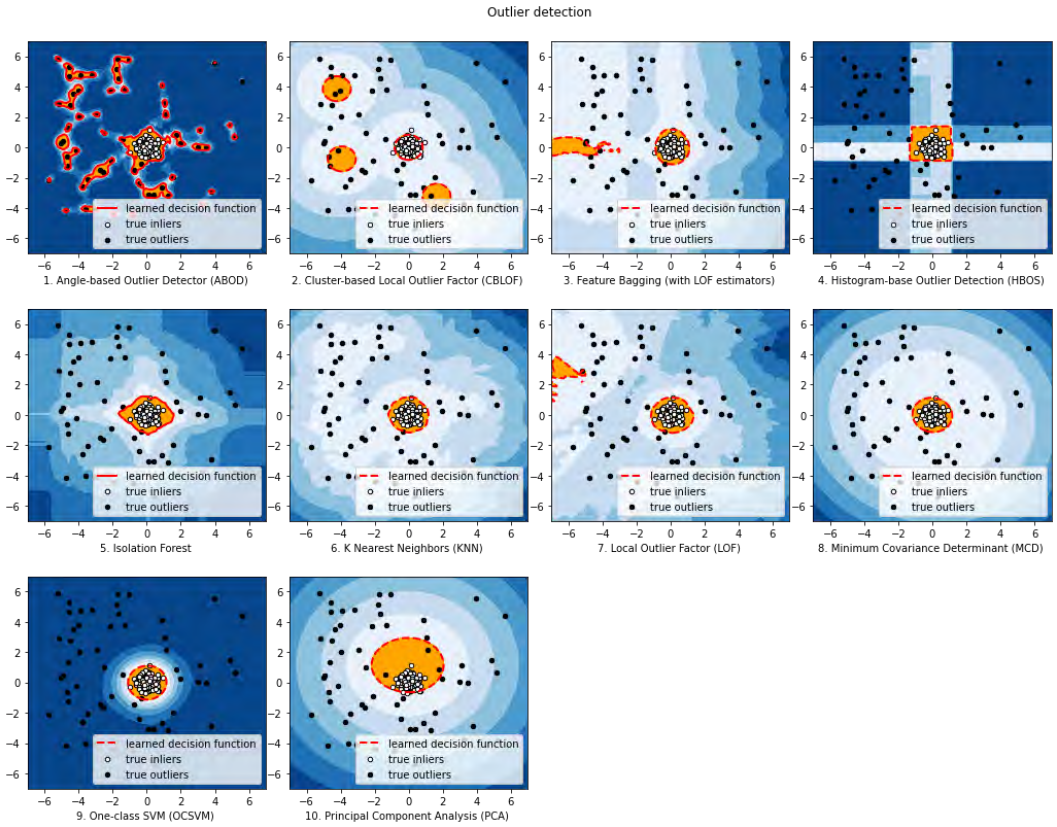

Figure 2.2: Decision boundaries generated by ten different outlier detection algorithms on a given dataset. Each algorithm employs unique assumptions and methodologies, resulting in varied boundary formations and highlighting the importance of selecting the most effective models for anomaly detection.

2.2.2 Unsupervised Feature Selection with Fixed SPEC

In ensemble-based unsupervised outlier detection, selecting the most relevant base models is critical to improving robustness, efficiency, and interpretability, especially in the absence of labeled data. Rather than applying feature selection to raw data, this methodology applies unsupervised feature selection to the confidence scores generated by each base model. This meta-learning approach filters out less relevant detectors, retaining only models that effectively distinguish outliers from inliers. By reducing noise and redundancy, the ensemble focuses on the most reliable detectors, enhancing accuracy and reducing complexity.

The feature selection process uses Spectral Feature Selection (SPEC), specifically the Fixed SPEC variant [66]. SPEC uses spectral graph theory to assess the relevance of the feature by assessing the alignment with the intrinsic structure of the data. In this context, the intrinsic structure reflects the clustering of data points into two primary groups: outliers and inliers.

Using Fixed SPEC, we assume that the data can be divided into two clusters ($k = 2$), aligned with the objective of distinguishing normal data points from anomalies. This assumption directly supports outlier detection tasks by preserving the local graph structure critical to identifying anomalous patterns. Techniques like the Laplacian score [73], which prioritizes features that maintain local structure, further validate the effectiveness of graph-based approaches in unsupervised tasks. A broader review of clustering methods [74] underscores the relevance of such techniques for tasks that require unsupervised structure preservation.

By dynamically selecting the most relevant base models, the integration of meta-learning with Fixed SPEC enhances the robustness and efficiency of the ensemble. This approach ensures that final predictions are informed by the most reliable detectors, which makes it well-suited for noisy and high-dimensional datasets.

2.2.2.1 Step 1: Graph Construction

The first step of the Fixed SPEC process involves constructing a similarity graph based on the outlier scores generated by the base detectors. Let

$$S \in \mathbb{R}^{n \times m}$$

represent the outlier score matrix, where n is the number of data points and m is the number of base detectors. Each row $s_i \in \mathbb{R}^m$ corresponds to the vector of outlier scores assigned to data point i.

To quantify the similarity between data points, we adopt the Radial Basis Function (RBF) kernel, a widely used measure in spectral clustering and manifold learning methods. The RBF kernel effectively captures local geometric relationships by assigning high similarity values to data points that have similar outlier score pat-

terns and rapidly decreasing similarity as their distance increases. Specifically, for each pair of data points i and j, we compute their similarity as follows:

$$W_{ij} = \exp\left(-\frac{\|s_i - s_j\|^2}{2\sigma^2}\right),$$

where:

- s_i, s_j are the outlier score vectors for data points i and j, respectively.
- σ is a scaling factor controlling the width of the kernel.

This computation yields the symmetric similarity matrix:

$$W \in \mathbb{R}^{n\times n},$$

where each element W_{ij} quantifies how similar two data points are in terms of their outlier scores.

Choice of Scaling Parameter (σ)

Selecting an appropriate value for σ is crucial for effective graph construction. Common heuristics include:

- Setting σ^2 as the median squared distance between all pairs of data points, ensuring balanced representation of local neighborhoods.
- Employing cross-validation or domain knowledge to optimize clustering performance or downstream analysis tasks.

Intuition and Theoretical Justification

The RBF kernel is particularly suitable for this scenario because it implicitly assumes that data points with similar outlier detection profiles lie close to one another on an underlying manifold structure. By encoding these local similarities into matrix W, we create a robust foundation for subsequent spectral analysis. Specifically, this similarity graph allows us to leverage spectral graph theory techniques—such as Laplacian eigenmaps—to uncover intrinsic connectivity patterns within the dataset.

In summary, this step transforms raw outlier detection outputs into a structured representation that captures both local and global relationships among data points. This structured representation forms the basis for computing meaningful spectral

embeddings in subsequent steps.

2.2.2.2 Step 2: Laplacian Matrix Calculation

Once the similarity graph is constructed, we proceed to compute the graph Laplacian, a fundamental operator in spectral graph theory that captures both local and global connectivity patterns within the graph. The Laplacian matrix provides a discrete analog of the continuous Laplace-Beltrami operator from differential geometry, effectively measuring how smoothly features vary across the underlying data manifold.

Formally, we first define the degree matrix D as a diagonal matrix whose entries represent the total connectivity (degree) of each node:

$$D_{ii} = \sum_{j=1}^{n} W_{ij}.$$

Intuitively, each diagonal entry D_{ii} quantifies how strongly node i is connected to all other nodes in the graph.

Using the degree matrix D, we construct the unnormalized Laplacian matrix L:

$$L = D - W.$$

The unnormalized Laplacian has several important properties:

- It is symmetric and positive semi-definite.
- Its smallest eigenvalue is always zero, with a corresponding eigenvector of constant entries.
- It encodes local smoothness: for any vector $f \in \mathbb{R}^n$, the quadratic form $f^T L f = \frac{1}{2}\sum_{i,j} W_{ij}(f_i - f_j)^2$ measures how smoothly f varies across connected nodes.

Relative Graph Structure

The concept of relative structure refers to analyzing connectivity patterns by considering each node's connection strengths relative to its own degree. By evaluating connections in this manner, we avoid scenarios where nodes with exceptionally high

absolute similarity values or degrees dominate the analysis. The Laplacian matrix inherently captures this relative structure because it measures deviations between actual connection strengths (given by W) and expected connectivity (represented by D).

To further emphasize relative connectivity and mitigate degree-related biases, we employ the normalized Laplacian:

$$L_{\text{norm}} = D^{-1/2} L D^{-1/2}.$$

This normalization scales each node's contribution according to its overall connectivity. As a result, spectral properties derived from L_{norm} reflect intrinsic structural patterns rather than being disproportionately influenced by highly connected nodes. The normalized Laplacian thus provides a balanced representation of the data structure suitable for robust spectral analysis in subsequent steps.

2.2.2.3 Step 3: Eigenvalue Decomposition and Clustering

With the normalized Laplacian L_{norm} computed, we proceed to perform eigenvalue decomposition, a central step in spectral clustering methods. Specifically, we solve the following eigenvalue problem:

$$L_{\text{norm}} v = \lambda v,$$

where λ denotes the eigenvalues and v the corresponding eigenvectors. Due to the symmetry and positive semi-definiteness of L_{norm}, all eigenvalues are real and non-negative.

Intuitively, each eigenvector represents a mode of variation or "signal" defined over the nodes of the graph, while its associated eigenvalue quantifies how smoothly this signal varies across connected nodes. A smaller eigenvalue indicates a smoother variation, meaning that connected nodes tend to have similar values in the corresponding eigenvector. Conversely, larger eigenvalues correspond to more rapidly oscillating patterns across the graph.

The smallest eigenvalue of L_{norm} is always $\lambda_1 = 0$, with an associated constant

eigenvector. The second smallest eigenvalue λ_2, known as the *Fiedler value*, and its corresponding eigenvector v_2, known as the *Fiedler vector*, are particularly significant. The Fiedler vector captures the most fundamental nontrivial mode of variation across the graph, effectively encoding its principal connectivity structure.

Spectral Clustering via Fiedler Vector

In practical terms, the components of the Fiedler vector v_2 provide a scalar embedding of nodes into a one-dimensional space that respects their relative connectivity patterns. Nodes that are strongly connected (relative to their degrees) will have similar values in v_2. Therefore, by applying a suitable threshold—such as partitioning nodes based on the sign or median value of their entries in v_2—we naturally obtain an optimal two-way partition of the graph.

This partition represents a spectral approximation to an optimal graph cut that minimizes edge weights between clusters while preserving internal cluster connectivity. Thus, spectral clustering using the Fiedler vector leverages intrinsic structural information encoded within the normalized Laplacian, enabling robust identification of meaningful clusters within complex data structures.

2.2.2.4 Step 4: Feature Ranking and Selection

In this final step, we assess each feature (i.e., each column s_j of the outlier score matrix S) based on how well it aligns with the intrinsic graph structure captured by the Fiedler vector v_2. The intuition behind this approach is that features which reflect the underlying connectivity patterns of the data manifold will exhibit variations similar to those encoded in v_2.

Formally, we quantify this alignment using cosine similarity. For each feature vector $s_j \in \mathbb{R}^n$, we compute its cosine similarity with the Fiedler vector v_2:

$$\text{cosine similarity}(v_2, s_j) = \frac{v_2 \cdot s_j}{\|v_2\|\|s_j\|}.$$

A higher absolute cosine similarity indicates stronger alignment between the feature's variation and the main mode of connectivity variation represented by v_2. Features with high cosine similarity thus better capture meaningful structural dif-

ferences within the data.

Relevance Score Definition

To incorporate both the alignment quality (cosine similarity) and the clarity of the spectral partition (captured by the Fiedler eigenvalue λ_2), we define a composite relevance score for each feature:

$$\text{score}_j = (2-\lambda_2) \cdot (\text{cosine similarity}(v_2, s_j))^2.$$

In this formulation:

- The factor $(2-\lambda_2)$ emphasizes features derived from graphs with clearer partitions. A smaller value of λ_2 indicates a more distinct two-cluster structure, thus increasing the relevance of aligned features.
- The squared cosine similarity term ensures that features strongly aligned with the intrinsic connectivity pattern receive significantly higher scores, while reducing sensitivity to minor variations.

Finally, we rank all features according to their computed relevance scores. Features receiving higher scores are considered more significant, as they effectively capture and represent meaningful structural patterns within the data. To construct a robust ensemble that prioritizes informative detectors and reduces noise or redundancy, we select and retain only the top 60% of features based on these scores. This selection procedure ensures that our final ensemble consists primarily of base detectors whose outputs align closely with the intrinsic manifold structure revealed through spectral analysis, thereby enhancing overall detection performance and interpretability.

2.2.2.5 Final Remarks

The fixed SPEC-based unsupervised feature selection process is integral to improving the robustness and interpretability of our ensemble method. By retaining only the most relevant models reduces susceptibility to noise, improves detection accuracy, and simplifies the ensemble by discarding redundant or less effective detectors. This streamlining enhances both efficiency and interpretability.

In addition to fixed SPEC, we evaluated several alternative unsupervised feature selection methods for comparative analysis:

- **Generic SPEC:** A general version of the SPEC framework that does not assume a fixed number of clusters [66].
- **Normalized Cut:** A graph-based method that minimizes cuts between clusters while maximizing within-cluster similarity [75].
- **Unsupervised Lasso:** A method that uses L1 norm regularization to perform simultaneous feature selection and clustering [76].
- **Weighted K-Means (WKMeans):** A variation of k-means that assigns weights to features based on their importance to clustering [77].

Our evaluations showed that fixed SPEC consistently outperformed these methods, achieving superior retention of informative base detectors while maintaining computational simplicity. By adapting the principles of variance-based feature selection [78] to the meta-learning context, where features represent outlier scores, a fixed SPEC ensures a more robust and efficient ensemble for unsupervised outlier detection across diverse datasets.

2.2.3 Ensemble Aggregation and Voting Mechanism

After selecting the most relevant base detectors through the unsupervised feature selection process, their outlier scores are aggregated to produce the final predictions. Ensemble aggregation enhances the reliability of outlier detection by balancing diverse outputs, maximizing accuracy and robustness, while minimizing the influence of noisy or weak models.

Several aggregation techniques were evaluated, each employing a distinct strategy for combining scores.

- **Average:** Computes the mean of outlier scores in all selected detectors. This method assumes equal contributions from all detectors, making it suitable when all models are expected to perform comparably.
- **Maximization:** Selects the highest outlier score among all detectors. This approach highlights extreme outlier signals, assuming that the most significant anomaly score is a reliable indicator, regardless of other scores.

- **Average of Maximum (AOM):** Divide detectors into subgroups, computes the maximum score within each subgroup, and averages these maximum scores. AOM balances extreme outlier signals with model diversity, capturing strong anomaly signals without over-reliance on any single detector.
- **Maximum of Average (MOA):** Divides detectors into subgroups, computes the average score within each subgroup, and selects the maximum of these averages. MOA emphasizes the consistency of scores within subgroups, favoring models with aligned predictions over those that produce extreme outliers.

Among these methods, AOM emerged as the most robust, effectively capturing extreme outlier signals while preserving model diversity. Its balanced approach makes it particularly resilient to noise and biases from individual detectors, ensuring reliable final predictions.

2.2.3.1 Average of Maximum (AOM) Strategy

The Average of Maximum (AOM) strategy was selected as the aggregation method for the final ensemble due to its ability to balance sensitivity to extreme scores with robustness against individual model variability. AOM divides the selected detectors into g subgroups, calculates the maximum outlier score within each subgroup, and averages these maximum values to produce the final prediction. This approach incorporates diverse perspectives while mitigating the impact of noisy models.

Given m detectors, the outlier score for a data point x_i is calculated as:

$$S_{\text{AOM}}(x_i) = \frac{1}{g} \sum_{j=1}^{g} \max_{d \in G_j} S_d(x_i) \tag{2.1}$$

where $S_d(x_i)$ is the outlier score of x_i produced by the dth detector, and G_j is the jth subgroup of detectors. The final score is the average of the maximum scores in all subgroups.

2.2.3.2 Advantages of AOM for Outlier Detection

The AOM strategy offers several key advantages for unsupervised outlier detection:

- **Balancing extreme and average scores:** By capturing the maximum score

within each subgroup and averaging across subgroups, AOM balances sensitivity to extreme outlier signals with stability, reducing false positives caused by over-reliance on a single detector.

- **Leveraging model diversity:** Subgrouping ensures that a range of models contributes to the final prediction, allowing the ensemble to detect complex outlier patterns that might be missed by individual models.
- **Mitigating noisy models:** By focusing on subgroup maxima rather than individual scores, AOM reduces the influence of noisy or weak models, improving the reliability of the ensemble across varied datasets.

This combination of sensitivity, robustness, and diversity makes AOM an ideal aggregation method for unsupervised outlier detection, ensuring accuracy and resilience in diverse and noisy data environments.

2.2.3.3 Summary

The AOM method was selected as the optimal ensemble aggregation technique due to its ability to balance the sensitivity to extreme outliers with the diversity of base models. By capturing both extreme and typical outlier scores across detectors, AOM enhances accuracy and robustness, particularly in noisy or high-dimensional datasets.

While alternative methods such as Average, Maximization, and MOA offered specific strengths, AOM consistently provided the best trade-off between robustness and precision. This choice finalizes the ensemble design, ensuring a reliable foundation for effective outlier detection.

2.3 Data and Experiments

This section outlines the datasets and experiments used to validate the proposed methodology. It provides details on the dataset sources, characteristics, and preprocessing steps, followed by a description of the experimental setup, evaluation metrics, and parameter configurations.

2.3.1 Datasets

The proposed methodology was validated using 17 benchmark datasets from the PyOD (Python Outlier Detection) library [79], a widely recognized resource in outlier detection research. These datasets span various domains, sample sizes, and outlier percentages, offering a comprehensive foundation for testing the robustness and adaptability of outlier detection methods. The use of standardized PyOD datasets ensures fair and consistent comparisons, minimizing external biases and reflecting the inherent capabilities of each model.

Table 2.1 summarizes the datasets, highlighting key attributes such as sample size, dimensionality, outlier percentage, and domain. These attributes contextualize the datasets' suitability for benchmarking outlier detection performance. All datasets and evaluation metrics are publicly available through the PyOD GitHub repository, facilitating reproducibility and comparative analysis in future research.

2.3.2 Experimental Setup

This subsection outlines the preprocessing steps, the training process, evaluation metrics, statistical analysis, and implementation details.

2.3.2.1 Preprocessing Steps

Each dataset was split into 60% for training and 40% for testing. The data were then standardized by subtracting the mean and dividing by the standard deviation for each feature. Standardization ensures consistency across datasets, enabling distance-based models like k-NN to operate effectively on comparable scales while maintaining compatibility with scale-invariant algorithms such as iForest. This preprocessing step enhances model performance and ensures reliable comparisons across diverse detection algorithms.

2.3.2.2 Training Process

The training process for the proposed methodology consists of four key steps.

1. **Training Base Outlier Detectors:** A diverse set of 10 unsupervised outlier detection algorithms, including k-Nearest Neighbors (k-NN), Local Out-

Table 2.1: Overview of datasets used for experimental validation, detailing sample size, dimensionality, outlier percentage, domain, and data type.

Dataset	#Samples	#Dimensions	%Outliers	Domain	Data Type
arrhythmia	452	274	14.60	Healthcare	Tabular
cardio	1831	21	9.61	Healthcare	Tabular
glass	214	9	4.21	Forensics	Tabular
ionosphere	351	33	35.90	Space Physics	Tabular
letter	1600	32	6.25	Computer Vision	Tabular
lympho	148	18	4.05	Healthcare	Tabular
mnist	7603	100	9.21	Computer Vision	Image
musk	3062	166	3.17	Chemistry	Tabular
optdigits	5216	64	2.88	Computer Vision	Tabular
pendigits	6870	16	2.27	Computer Vision	Tabular
pima	768	8	34.90	Healthcare	Tabular
satellite	6435	36	31.64	Remote Sensing	Tabular
satimage-2	5803	36	1.22	Remote Sensing	Tabular
shuttle	49097	9	7.15	Aerospace	Tabular
vertebral	240	6	12.50	Healthcare	Tabular
vowels	1456	12	3.43	Linguistics	Tabular
wbc	378	30	5.56	Healthcare	Tabular

lier Factor (LOF), Cluster Based Local Outlier Factor (CBLOF), Histogram Based Outlier Score (HBOS) and Isolation Forest (iForest), was trained on the training data. This diversity enables the ensemble to capture a broad range of anomaly types by leveraging different perspectives in outlier detection.

2. **Outlier Score Generation:** Each base model generated outlier scores for all training data points. These scores served as features, providing multiple representations of data point "outlierness" based on different detection strategies.

3. **Unsupervised Feature Selection with Fixed SPEC:** The Fixed SPEC algorithm ranked the base detectors according to relevance, retaining the top 60% that aligned the most closely with the data structure. This process improved the ensemble's focus on informative models, enhancing detection accuracy and robustness.

4. **Ensemble Aggregation:** Outlier scores from selected detectors were aggregated using the Average of Maximum (AOM) voting strategy. Subgroups of detectors were formed and the maximum score within each subgroup was computed. The final prediction was derived by averaging these maximum scores, ensuring a balance between diversity and robustness.

2.3.2.3 Evaluation Metrics

To ensure a fair and comprehensive comparison of the proposed methodology against other outlier detection models, we used two key metrics: AUC-ROC and Precision.

Area Under the Receiver Operating Characteristic Curve (AUC-ROC)

AUC-ROC measures a model's ability to distinguish between classes by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) across various thresholds. These rates are defined as follows:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2.2}$$

where:

- TP: True Positives (correctly identified outliers)
- FP: False Positives (inliers misclassified as outliers)
- FN: False Negatives (outliers misclassified as inliers)
- TN: True Negatives (correctly identified inliers)

The AUC score ranges from 0.5 (random guessing) to 1.0 (perfect discrimination). It is particularly suited for scenarios where the trade-off between true and false positives is critical, such as fraud detection and network security.

Precision

Precision evaluates the proportion of correctly identified outliers among all instances flagged as outliers:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.3}$$

This metric is essential in contexts where false positives have significant costs,

such as financial fraud detection, where high precision ensures that flagged transactions are more likely to be genuine anomalies, reducing unnecessary investigations.

AUC-ROC and Precision together provide a balanced framework for evaluating both the sensitivity and reliability of outlier detection models, ensuring robust performance in diverse application settings.

2.3.2.4 Statistical Analysis: Friedman Test

To determine whether observed performance differences across outlier detection models are statistically significant, we used the non-parametric Friedman test. This test is well-suited for comparing multiple algorithms on multiple datasets, particularly in machine learning contexts where performance varies non-linearly, and parametric assumptions may not hold.

For k algorithms evaluated on N datasets, the Friedman test ranks the algorithms for each dataset j from 1 (best) to k (worst). The average rank R_i of each algorithm i is calculated as follows:

$$R_i = \frac{1}{N} \sum_{j=1}^{N} r_{ij}$$

where r_{ij} is the rank of the ith algorithm in the jth dataset.

The Friedman test statistic χ_F^2 is computed as:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left(\sum_{i=1}^{k} R_i^2 - \frac{k(k+1)^2}{4} \right)$$

This statistic follows a chi-square distribution with $k-1$ degrees of freedom, allowing us to test whether the rank differences between datasets are statistically meaningful.

The Friedman test offers a robust statistical framework for evaluating whether performance differences between algorithms are significant or due to random variation. This is particularly valuable in machine learning research and practice, where algorithm comparisons across benchmarks are common. By applying the Friedman test, researchers and practitioners can confidently assess performance disparities,

helping in model selection, and ensuring that decisions are based on statistically meaningful insights, which is critical for real-world applications with significant outcomes.

2.3.2.5 Experiments

The experimental framework was designed to assess the effectiveness of the proposed methodology and assess the individual contributions of its feature selection and ensemble voting components. Three main experiments were conducted: comparison of the baseline model, comparison of the feature selection method, and comparison of the ensemble voting method.

Experiment 1: Overall Performance Evaluation

This experiment compared the proposed methodology with ten established outlier detection techniques to validate its state-of-the-art performance. Baseline methods included k nearest neighbors (k-NN), local outlier factor (LOF), cluster-based LOF (CBLOF), histogram-based outlier score (HBOS), one-class SVM (OCSVM), minimum covariance determinant (MCD), isolation forest (iForest), principal component analysis (PCA), angle-based outlier detection (ABOD), and feature bagging (FB).

Experiment 2: Impact of Feature Selection

The second experiment isolated the effect of the proposed Fixed SPEC feature selection method by comparing it with alternatives such as fixed clustering, arbitrary clustering, normalized cut, and weighted k-means. The base models and the aggregation strategy were kept constant to accurately measure the impact of feature selection on detection performance.

Experiment 3: Impact of Ensemble Voting

This experiment assessed the influence of the proposed aggregation strategy, the 'average of maximum' (AOM), by comparing it against alternative voting methods, including simple average, maximization, and maximum of average (MOA). The feature selection method and base models were fixed to isolate the impact of each voting technique on the final predictions of the ensemble.

Table 2.2: AUC-ROC: Raw results per model and dataset (average of 5 trials).

Data	#Rows	#Dims	%Outliers	ABOD	CBLOF	FB	HBOS	IForest	KNN	LOF	MCD	OCSVM	PCA	AOM	PROPOSED
arrhythmia	452	274	14.6018	0.75478	0.76042	0.76624	0.83304	0.83536	0.76834	0.767	0.786	0.777	0.7779	0.79932	0.80636
cardio	1831	21	9.6122	0.55894	0.82858	0.51898	0.84824	0.94268	0.70402	0.50432	0.87412	0.94658	0.96004	0.9136	0.91544
glass	214	9	4.2056	0.74278	0.86444	0.7806	0.6897	0.7364	0.81004	0.82102	0.73902	0.62024	0.6295	0.74618	0.71386
ionosphere	351	33	35.8974	0.91228	0.902	0.89394	0.54206	0.84296	0.92442	0.89956	0.94716	0.85314	0.80264	0.8687	0.86776
letter	1600	32	6.25	0.86142	0.75734	0.86804	0.58572	0.59716	0.85534	0.85218	0.77182	0.5809	0.49494	0.79738	0.78382
lympho	148	18	4.0541	0.9043	0.96702	0.96314	0.99784	0.9913	0.96264	0.96092	0.9045	0.96264	0.98046	0.97364	0.98664
mnist	7603	100	9.2069	0.78148	0.84176	0.7292	0.56444	0.78834	0.84392	0.72004	0.84854	0.83976	0.83872	0.8291	0.81276
musk	3062	166	3.1679	0.10668	1	0.53836	0.99994	0.99944	0.7665	0.54098	0.99974	1	0.99998	0.99972	0.99522
optdigits	5216	64	2.8758	0.4515	0.76966	0.4691	0.87552	0.61316	0.37698	0.46556	0.3584	0.49338	0.5019	0.68606	0.70976
pendigits	6870	16	2.2707	0.69892	0.96238	0.48352	0.92942	0.94228	0.75506	0.49278	0.82788	0.92926	0.93426	0.92598	0.9094
pima	768	8	34.8958	0.67502	0.6728	0.63302	0.70656	0.65712	0.7175	0.64198	0.6979	0.63014	0.6485	0.68686	0.69372
satellite	6435	36	31.6395	0.56742	0.72168	0.55492	0.74798	0.69672	0.67996	0.55398	0.79804	0.64634	0.58926	0.72258	0.7279
satimage-2	5803	36	1.2235	0.855	0.99816	0.49052	0.9723	0.9904	0.95222	0.48928	0.99458	0.99548	0.97168	0.99648	0.9914
shuttle	49097	9	7.1511	0.62174	0.62754	0.5201	0.98438	0.99692	0.64868	0.53478	0.98996	0.99154	0.98958	0.99576	0.99396
vertebral	240	6	12.5	0.33178	0.38306	0.35728	0.31012	0.41318	0.35792	0.36848	0.36968	0.45958	0.3993	0.34466	0.3712
vowels	1456	12	3.4341	0.95528	0.91872	0.93292	0.69868	0.72978	0.96316	0.9298	0.69494	0.77312	0.60994	0.9016	0.93778
wbc	378	30	5.5556	0.91486	0.92016	0.94198	0.95988	0.94378	0.94226	0.93542	0.90236	0.93878	0.93096	0.94576	0.94966

2.3.2.6 Implementation Details

All experiments were implemented using the PyOD Python library [79]. Each experiment was repeated five times to ensure the consistency of the results, with the final scores representing the average performance among these runs for each dataset.

2.4 Results and Discussion

This section presents the findings of our experiments, evaluates the proposed methodology using AUC-ROC and precision metrics, and compares its performance with baseline and alternative methods. The raw results of the first experiment are summarized in Tables 2.2 and 2.3 for AUC-ROC and precision, respectively.

2.4.1 Experiment 1: Performance of the Proposed Methodology

The first experiment evaluated the overall effectiveness of the proposed methodology in unsupervised outlier detection by comparing its performance to a variety of established algorithms. This evaluation focused on two key metrics: AUC-ROC and precision, both of which are critical indicators of the robustness and accuracy of an outlier detection system.

Table 2.3: Precision: Raw results per model and dataset (average of 5 trials).

Data	#Rows	#Dims	Outlier Perc	ABOD	CBLOF	FB	HBOS	IForest	KNN	LOF	MCD	OCSVM	PCA	AOM	PROPOSED
arrhythmia	452	274	14.6018	0.34726	0.42538	0.4468	0.53914	0.53226	0.4468	0.43942	0.3835	0.4537	0.4468	0.48252	0.51108
cardio	1831	21	9.6122	0.19314	0.49558	0.1307	0.46098	0.51398	0.30234	0.1367	0.42454	0.55406	0.65912	0.41988	0.44812
glass	214	9	4.2056	0.23	0.19	0.27	0.04	0.19	0.19	0.27	0	0.19	0.19	0.19	0.19
ionosphere	351	33	35.8974	0.82946	0.8031	0.7551	0.34232	0.66184	0.86012	0.7551	0.86674	0.72184	0.59522	0.72576	0.75152
letter	1600	32	6.25	0.29908	0.19576	0.36988	0.1053	0.07544	0.30538	0.32488	0.11414	0.11834	0.05386	0.22008	0.21428
lympho	148	18	4.0541	0.34	0.56	0.56	0.88	0.76	0.56	0.56	0.56	0.56	0.68	0.56	0.74
mnist	7603	100	9.2069	0.35008	0.40298	0.36356	0.1167	0.28742	0.41504	0.34344	0.35448	0.3772	0.3724	0.37578	0.35494
musk	3062	166	3.1679	0.04908	1	0.19502	0.97548	0.92588	0.2618	0.16998	0.95502	1	0.99	0.96214	0.79334
optdigits	5216	64	2.8758	0.0236	0	0.03268	0.23424	0.00966	0	0.02904	0	0	0	0.00322	0.01612
pendigits	6870	16	2.2707	0.0596	0.32224	0.06194	0.2796	0.31412	0.07138	0.0585	0.06952	0.30956	0.33176	0.26516	0.2249
pima	768	8	34.8958	0.50892	0.47066	0.435	0.5085	0.48504	0.51154	0.45544	0.4928	0.46252	0.50084	0.48514	0.51868
satellite	6435	36	31.6395	0.39006	0.55224	0.40148	0.55862	0.58046	0.49674	0.40076	0.6786	0.5278	0.4679	0.59938	0.59226
satimage-2	5803	36	1.2235	0.24624	0.88134	0.06712	0.6289	0.85068	0.36758	0.07246	0.5631	0.90354	0.82048	0.73912	0.52934
shuttle	49097	9	7.1511	0.19844	0.23528	0.11194	0.97682	0.9357	0.22658	0.15128	0.74876	0.95396	0.94942	0.94168	0.91272
vertebral	240	6	12.5	0.01538	0.01538	0.03076	0.01538	0.03076	0	0.03076	0	0.01538	0	0.01538	0.01538
vowels	1456	12	3.4341	0.4888	0.35528	0.3205	0.14918	0.19304	0.52408	0.35584	0.01112	0.3174	0.13548	0.29576	0.39978
wbc	378	30	5.5556	0.255	0.535	0.5	0.695	0.475	0.475	0.415	0.39	0.475	0.535	0.535	0.495

Table 2.4: Friedman test rankings comparing the proposed methodology against other outlier detection algorithms based on AUC-ROC and Precision metrics.

AUC-ROC

Rank	Algorithm
4.8823	Proposed methodology
5.0588	Average of maximum
5.1470	Cluster-based local outlier factor
5.3529	Isolation forest
6.0882	K Nearest neighbours (KNN)
6.1176	Histogram-base outlier detection (HBOS)
6.1764	Minimum covariance determinant (MCD)
6.4705	One-class SVM (OCSVM)
6.9411	Principal component analysis (PCA)
8.4705	Feature bagging
8.5882	Angle-based outlier detector (ABOD)
8.7058	Local outlier factor (LOF)

Precision

Rank	Algorithm
5.44117	Proposed methodology
5.58823	Cluster-based local outlier factor
5.61764	Histogram-base outlier detection (HBOS)
5.73529	Average of maximum
5.82352	One-class SVM (OCSVM)
5.94117	Isolation forest
6.29411	Principal component analysis (PCA)
6.58823	K nearest neighbours (KNN)
7.17647	Feature bagging
7.82352	Local outlier factor (LOF)
7.97058	Angle-based outlier detector (ABOD)
8	Minimum covariance determinant (MCD)

The proposed methodology outperformed competing approaches, achieving the highest rankings in both AUC-ROC and precision metrics, as shown in Table 2.4. These results demonstrate the robustness of the methodology and its capacity to improve detection accuracy through the integration of feature selection.

A comparison with the baseline "average of maximum" ensemble without feature selection shows its moderately strong performance (second in AUC-ROC,

Table 2.5: Selection frequency of each outlier detection method by the unsupervised feature selection algorithm, expressed as a percentage. The values indicate the proportion of times where each method was retained as part of the ensemble.

Percentage	Algorithm
0.80	Cluster-based local outlier factor
0.76	Isolation forest
0.73	k-Nearest Neighbors (k-NN)
0.71	Histogram-based outlier detection (HBOS)
0.65	Minimum covariance determinant (MCD)
0.59	One-class SVM (OCSVM)
0.53	Principal component analysis (PCA)
0.45	Feature bagging
0.40	Angle-based outlier detector (ABOD)
0.39	Local outlier factor (LOF)

fourth in precision). However, the proposed methodology outperformed this baseline, underscoring the critical role of feature selection in filtering less effective models and improving ensemble performance.

Detailed rankings in Table 2.4 reveal that classic methods like Isolation Forest and Cluster-Based Local Outlier Factor (CBLOF) consistently ranked among the top five, showcasing their adaptability to diverse datasets. In contrast, methods such as the Local Outlier Factor (LOF) and Angle-Based Outlier Detector (ABOD) were ranked lower, suggesting that their effectiveness is more dataset-specific and less generalizable.

Table 2.5 highlights the selection frequency of individual algorithms, offering insight into their contributions to the ensemble. CBLOF and Isolation Forest were selected most frequently, indicating their reliability and ability to capture complex outlier patterns across datasets. In contrast, models such as LOF and ABOD had lower selection rates, reaffirming their context-dependent performance. These findings emphasize the value of feature selection in enhancing the stability and effectiveness of the ensemble, particularly in challenging scenarios involving high-dimensional or noisy data.

Table 2.6: Friedman test rankings comparing the performance of different feature selection techniques within the proposed methodology based on AUC-ROC and Precision metrics.

AUC-ROC

Rank	Algorithm
3.05882	Fixed clustering
3.32352	Normalised cut
3.5	Lasso
3.88235	Weighted k-means
4.11764	Arbitrary clustering

Precision

Rank	Algorithm
3.0294	Fixed clustering
3.1764	Normalised cut
3.6764	Lasso
3.8235	Weighted k-means
4	Arbitrary clustering

2.4.2 Experiment 2: Impact of Feature Selection Methods

This experiment evaluated the impact of feature selection methods, with Fixed SPEC outperforming alternatives by effectively prioritizing informative base detectors. Keeping the ensemble aggregation strategy constant, the experiment isolated the role of feature selection in enhancing performance.

As shown in Table 2.6, the Friedman test rankings confirm that Fixed SPEC consistently achieved the highest scores in both metrics, highlighting its ability to enhance ensemble accuracy and robustness by retaining the most relevant detectors.

The comparison included alternative methods such as normalized cut, lasso, and weighted k-means, each employing different strategies for feature selection. These alternatives provided valuable insights into the advantages of Fixed SPEC:

- **Fixed Clustering**: Ranked second to Fixed SPEC, this method predefines groups for selection, achieving reasonable results but lacking the adaptability of Fixed SPEC to dataset-specific variations.
- **Normalized Cut**: The purpose is to partition data by minimizing the cuts in the cluster boundary. Although competitive, it struggles to match the fine-grained adaptability of Fixed SPEC in high-dimensional datasets.
- **Lasso**: Employs L1 regularization for sparsity but ranked midrange, limited by its reliance on linear assumptions, which can underperform in non-linear or complex data structures common in outlier detection.
- **Weighted K-means**: The least effective performance indicates that its

clustering-based approach does not align well with the structure of outlier scores in diverse datasets.

The superior performance of Fixed SPEC underscores its ability to preserve intrinsic data structures while optimizing feature relevance in unsupervised contexts. By aligning feature selection with data clusters, Fixed SPEC enhances the ensemble's capacity to distinguish subtle outlier patterns from normal instances. This adaptability ensures consistent performance across varied datasets, making it ideal for real-world applications where outlier characteristics differ widely.

These findings highlight the critical role of feature selection in unsupervised learning, emphasizing that the choice of method can significantly influence the robustness and accuracy of the final detection system.

2.4.3 Experiment 3: Measuring the Impact of the Proposed Ensemble Voting Method

This experiment evaluated the effect of different ensemble voting strategies on outlier detection performance, focusing on balancing the capture of extreme outlier signals with prediction robustness. The Average of Maximum (AOM) technique was compared with alternative aggregation methods, including simple average, maximum, and maximum of average, to assess the influence of the aggregation strategy on overall performance:

- **Simple Average**: Although straightforward, this method assigns equal weight to all models, potentially diluting the impact of the most informative detectors. Its poor performance in both the ROC and precision rankings reflects its inability to prioritize high-performing models.
- **Maximisation**: This approach relies exclusively on the highest outlier score from any model. Although effective in identifying extreme anomalies, it risks overfitting and high false positives by disproportionately emphasizing single-model predictions, particularly if those predictions themselves are outliers.
- **Maximum of Average (MOA)**: MOA averages scores within subgroups and selects the maximum of these averages, offering a middle ground between simple averaging and maximization. However, it falls short of AOM in overall

Table 2.7: Friedman test rankings comparing the performance of different ensemble voting aggregation techniques within the proposed methodology based on AUC-ROC and Precision metrics.

AUC-ROC

Rank	Algorithm
1.82352	Average of maximum
2.52941	Maximisation
2.64705	Maximum of average
3	Average

Precision

Rank	Algorithm
2.17647	Average of maximum
2.35294	Maximum of average
2.41176	Maximisation
3.05882	Average

effectiveness, probably because of its inability to capture extreme signals as effectively.

The results of the Friedman test, shown in Table 2.7, demonstrate that AOM achieved the highest ranking in both the AUC-ROC and the precision metrics. This top performance underscores the strength of AOM in selectively amplifying extreme outlier scores while maintaining model diversity, resulting in more consistent and reliable detection outcomes.

The superior performance of AOM lies in its methodological advantage: dividing the ensemble into subgroups, computing the maximum score within each subgroup, and averaging these maxima. This process balances the contributions of strong models while preventing any single model's extreme score from dominating. As a result, AOM achieves a robust balance of accuracy and reliability, making it highly suitable for real-world applications in fields such as finance, healthcare, and cybersecurity, where accurate and reliable outlier detection is critical.

These findings highlight the importance of selecting an appropriate ensemble voting strategy in unsupervised learning, with AOM providing a robust and practical solution to integrate various outlier detection models.

2.5 Summary and Next Steps

This chapter presented a novel feature selection-based meta-learning framework for unsupervised outlier detection, designed to enhance model selection and improve overall performance. By integrating spectral feature selection and strategic aggregation techniques, the framework systematically evaluates and prioritizes the most ef-

fective detection algorithms, building robust ensembles that address key challenges in unsupervised outlier detection. The feature selection process also acts as a model selection mechanism, filtering out less effective detectors and ensuring that only the most relevant models contribute to the final predictions. This dual capability improves reliability and robustness in diverse data sets.

2.5.1 Key Results and Findings

The framework combines multiple outlier detection models, leveraging their strengths while filtering out less effective detectors using the Fixed SPEC feature selection process. This systematic approach improves both accuracy and robustness:

- Consistently outperformed standard techniques in 17 datasets, focusing on the most effective models.
- Achieved the highest rank in the non-parametric Friedman test for both AUC-ROC and precision, validating its state-of-the-art performance.
- Optimized the model selection process through spectral feature selection, reducing noise and aligning with the intrinsic data structure.
- Balanced extreme outlier signals and model diversity using the Average of Maximum (AOM) aggregation strategy, enhancing prediction robustness.

2.5.2 Real-World Applications and the Need for Interpretability

This framework addresses practical challenges in domains that involve large, noisy, or complex datasets. For example, in fraud detection, it can help minimize false positives by leveraging the most relevant models, while in healthcare, it aids in early anomaly detection to improve patient outcomes.

Despite its strengths in robustness and model selection, real-world applications often demand greater transparency and interpretability. High-stakes domains, such as healthcare and finance, require stakeholders to understand the reasoning behind predictions, particularly when ethical considerations and accountability are paramount. The complexity of the framework, which involves multiple detection algorithms and feature selection processes, can obscure the decision-making process,

posing challenges to transparency.

As AI systems become increasingly integral to critical applications, the demand for interpretability grows, not only to build user trust, but also to meet regulatory and ethical standards. Chapter 3 transitions to this critical theme, introducing a taxonomy of interpretability methods to improve transparency in machine learning systems. This taxonomy bridges the gap between performance and explainability, aligning machine learning systems with the practical and ethical requirements of real-world deployment.

Chapter 3

A Taxonomy for Machine Learning Interpretability Methods

Artificial intelligence (AI) has transitioned from a theoretical concept to a transformative force across diverse domains, driven by advances in computational power, learning algorithms, and access to vast datasets. Since the advent of deep learning in 2012, machine learning (ML) models have achieved superhuman performance in tasks such as image recognition and natural language processing, enabling applications in fields such as retail [80], finance [81], and healthcare [82–84].

However, as ML models become increasingly complex, understanding their decision-making processes has become a critical challenge. Although models such as deep neural networks and ensemble methods deliver state-of-the-art performance, they often function as "black boxes" [85–87]. Simpler "white-box" models, such as linear regression [88] and decision trees [89], offer transparency, but often lack predictive power. Balancing performance with interpretability has become a key requirement for AI systems, particularly in high-stakes applications.

Explainable Artificial Intelligence (XAI) addresses this challenge by improving model transparency, bridging the gap between performance and interpretability [90]. Public interest in interpretability has grown significantly, as reflected in Google Trends data (Figure 3.1), which shows a sharp increase in searches for "Explainable AI" from 2015 to 2024. This rise underscores the critical importance of interpretability in building trustworthy AI systems.

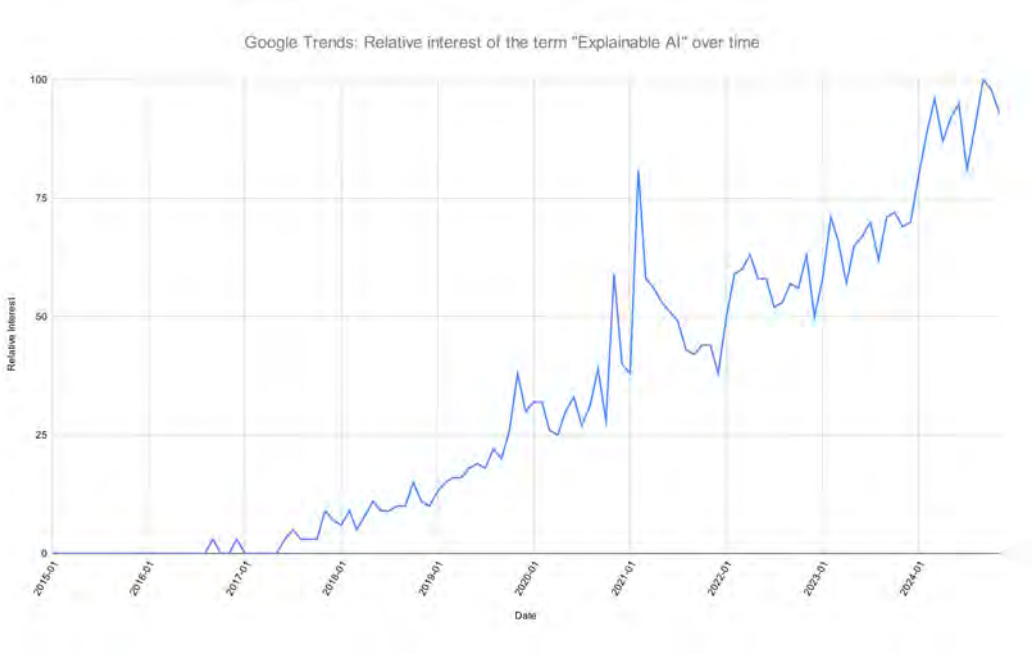

Figure 3.1: Google Trends: Relative Interest (maximum value of 100) for the term "Explainable AI" from 2015 to 2024.

In response to the demand for interpretability, this chapter presents a taxonomy of machine learning interpretability methods, offering a structured overview of this rapidly evolving field. The primary contributions of this chapter are as follows.

1. **Development of a New Taxonomy of Interpretability Methods:** A systematic framework categorizing interpretability techniques by functional approach and application domain.

2. **Extensive Review of Interpretability Methods:** A detailed analysis of current techniques, offering insights into their strengths and limitations.

3. **Comparative Analysis of Methods:** Practical guidance on selecting appropriate methods, supported by examples and recommendations.

4. **Guidance for Application and Resources:** Links to programming implementations and additional resources to support integration into workflows, provided in Appendix A.

By bridging theoretical advances and practical applications, this chapter empowers researchers and practitioners to develop interpretable ML models that are

reliable, transparent, and effective.

The remainder of this chapter is organized as follows. Section 3.1 reviews the foundational concepts and recent classification efforts in interpretability. Section 3.2 introduces our taxonomy, categorizing methods by functional approach and application domain. Section 3.3 presents a comparative analysis with practical recommendations. Section 3.3 explores the broader implications of interpretability in ethically sensitive applications. Finally, Section 3.4 concludes the chapter, summarizing the findings and setting the stage for the next chapter.

3.1 Key Concepts and Related Work

3.1.1 Explainability vs. Interpretability

Although "interpretability" and "explainability" are often used interchangeably, researchers have sought to clarify the distinctions between these terms. Both concepts are closely related, but lack precise mathematical definitions or standardized metrics, as highlighted in the works by Lipton [91], Doshi-Velez and Kim [92], and Gilpin et al. [93]. These terms, along with others such as "comprehensibility," remain informally defined [94]. Doshi-Velez and Kim [92] define interpretability as "the ability to explain or present in understandable terms to a human," while Miller [95] emphasizes interpretability as the extent to which a human can understand the cause of a decision. However, these definitions, while accessible, still lack mathematical rigor [94].

Interpretability focuses on allowing users to intuitively grasp the relationships between inputs and outputs of a model [94]. For example, in an image recognition system, interpretability would involve understanding why a specific visual pattern in the input led to a particular classification. Explainability, on the other hand, delves deeper into the internal mechanisms and decision-making processes of the model, offering transparency into how predictions are derived.

It is important to note that a model can be interpretable without being fully explainable. Users may recognize patterns in the model's behavior without understanding the underlying processes. In contrast, explainability often requires insight

into the inner workings of the model, as argued by Gilpin et al. [93]. This distinction underscores that interpretability alone may not suffice for trustworthy AI systems, particularly in high-stakes applications. Doshi-Velez and Kim [92] consider interpretability as a broader concept that includes explainability, a perspective that this study adopts to emphasize the importance of both.

3.1.2 Evaluation of Machine Learning Interpretability

Doshi-Velez and Kim [92] categorize interpretability evaluation methods into three main types: application-based, human-based, and functionally grounded approaches, each with distinct goals and trade-offs.

Application-Based Evaluation This approach assesses how effectively an interpretability method supports specific tasks or applications, focusing on benefits to end users or domain experts. For instance, it evaluates whether interpretability aids in identifying errors or reducing bias in decision-making within well-defined contexts.

Human-Based Evaluation While similar to application-based methods, human-based evaluation involves general users rather than domain experts and emphasizes abstract interpretability over task-specific utility. For example, evaluators might compare model explanations for an input and select the one they find most understandable.

Functionally-Grounded Evaluation This approach uses mathematical definitions and formal criteria to assess interpretability without human testing. It is particularly useful when human evaluation is impractical due to ethical concerns or early development stages of a model. While functionally grounded evaluations can refine models post-human testing, defining appropriate metrics and criteria remains a significant challenge in interpretability research.

Each of these methods addresses different aspects of interpretability, highlighting the need for tailored evaluation strategies based on the application context and stage of model development.

3.1.3 Related Work

Defining interpretability and explainability remains challenging, but various works have made progress toward formalization, including notable contributions by Doshi-Velez and Kim [92] and Lipton [91].

Gilpin et al. [93] sought to clarify interpretability in the context of machine learning, particularly deep learning, proposing a taxonomy for interpretability methods in neural networks. Their classification includes three categories: methods that simulate data processing to link inputs and outputs, approaches that interpret internal data representations, and self-explanatory models that offer inherent transparency. They emphasize that while progress has been made in explaining deep networks, the combination of different explanatory techniques could enhance interpretability.

Adadi and Berrada [94] conducted an extensive review of the literature, analyzing 381 studies from 2004 to 2018, and categorized explainable AI (XAI) research into four main dimensions. They argue for greater formalization and stronger human-machine interaction in XAI, noting a focus on explainability in modeling alone. They advocate for expanding explainability into other areas of machine learning and suggest combining existing methods to foster deeper insights.

Guidotti et al. [96] also classified explainability techniques, organizing them based on the problems each technique addresses. Their categories include explaining black-box models, inspecting them, interpreting outcomes, and creating inherently transparent black-box models. They further developed a taxonomy considering factors such as the type of explanation model, input data, problem type, and the specific black-box model examined. Like other studies, they highlight a lack of formal evaluation metrics and note that most black-box explanations struggle with models based on latent or unknown features. They also identify the need for more interpretability techniques in recommender systems and suggest developing models derived from explanations directly.

Murdoch et al. [97] responded to the lack of formal interpretability metrics with a 2019 survey introducing the Predictive, Descriptive, Relevant (PDR) frame-

work. This framework evaluates interpretability methods using three metrics: predictive accuracy, descriptive accuracy, and relevancy. They explore both transparent models and post-hoc interpretability, asserting that combining post-hoc explanations with transparent models can sometimes be the optimal approach, as it improves both usability and predictive accuracy.

A more recent work by Arrieta et al. [98] offers a different organization by initially distinguishing between transparent and post hoc methods, then categorizing subtypes. Due to the large number of deep learning interpretability methods, they also proposed a specialized taxonomy for this field. Their categories include methods that explain network processing, representation, producing systems, and hybrid approaches that blend transparency with black-box methods. They further introduce Responsible AI, outlining guidelines for implementing AI responsibly in organizations.

3.2 Proposed Taxonomy

Machine learning interpretability methods can be classified through multiple perspectives, such as algorithm compatibility, explanation scale, and data type. These diverse categorizations ensure that practitioners can identify the most suitable technique for specific problem requirements.

One primary distinction is between model-specific methods, which cater to particular algorithm families, and model-agnostic methods, which apply across diverse models. Another key differentiation lies in scale: local methods focus on individual instance explanations, while global methods address overall model behavior. Additionally, interpretability techniques vary by the type of data they target, such as tabular, image, or text formats. Figure 3.2 illustrates these facets in a mind map, providing a structured overview for selecting appropriate methods.

This taxonomy further categorizes interpretability methods into four primary groups:

- Methods for explaining complex black-box models.
- Techniques for building transparent (white-box) models.

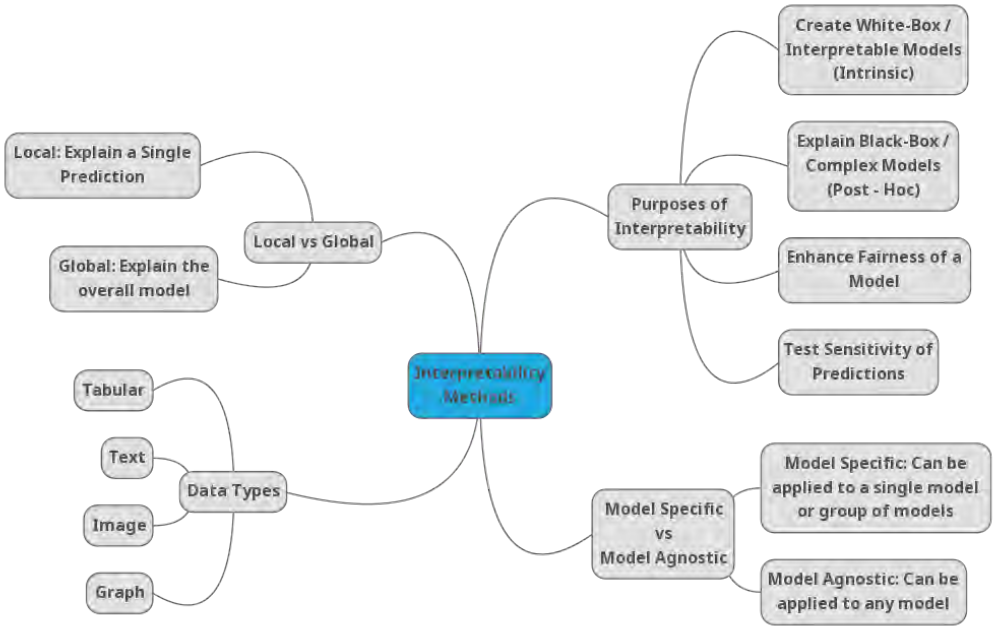

Figure 3.2: Taxonomy mind-map of Machine Learning Interpretability Techniques.

- Approaches to promote fairness and reduce discrimination.
- Methods for analyzing the sensitivity of model predictions.

By emphasizing the purpose and mechanism of each method, this taxonomy provides a practical framework for selecting interpretability techniques tailored to specific goals and data contexts.

3.2.1 Interpretability Methods to Explain Black-Box Models

This category includes post-hoc interpretability techniques designed to elucidate predictions from complex, pre-trained black-box models. These methods aim to provide insights into the decision-making processes of models such as deep neural networks without altering their structure. Interpretability methods are divided into two subcategories: those tailored for deep learning models (3.2.1.1) and those applicable to any black-box model (3.2.1.2). Tables 3.1 and 3.2 summarize these methods.

3.2.1.1 Interpretability Methods for Deep Learning Models

Given the prominence of deep learning in AI, significant research has focused on interpretability techniques, particularly for image-based tasks. These methods aim to analyze and visualize the features driving the predictions of the deep learning model using techniques such as saliency maps, gradients, and relevance scores.

Table 3.1 categorizes these methods by interpretative approach (e.g. post hoc techniques), scope (local or global), model specificity (model-specific or model-agnostic), supported data types (e.g. image, text, tabular) and available tools. This comprehensive overview highlights the depth and evolution of interpretability techniques in deep learning research.

For practical implementation, the Appendix Table A.1 lists the corresponding tools and repositories, serving as a hands-on resource to implement the methods discussed.

- **Gradient-Based Saliency Techniques:**
  - **Gradients**: This fundamental method calculates the gradient of the output with respect to the input features, identifying the parts of the image most influential to a given class [99].
  - **Integrated Gradients**: An extension of the basic gradient method, Integrated Gradients addresses some of the limitations of simple gradients by ensuring completeness, meaning that the feature attributions sum up to the difference between the output and the baseline prediction [100].
  - **SmoothGrad**: Designed to reduce noise in gradient-based saliency maps, SmoothGrad applies small random perturbations to the input and averages the resulting saliency maps, producing sharper and more coherent visualizations [101].
- **Decomposition-Based and Attribution Methods:**
  - **Deep Learning Important FeaTures (DeepLIFT)**: By comparing the activations of each neuron with a reference, DeepLIFT assigns importance scores to inputs, which can reveal dependencies missed by gradient-based methods alone [102].

  - **Layer-wise Relevance Propagation (LRP)**: LRP decomposes the output prediction layer by layer, assigning relevance scores through the network to the input layer, providing insight into which features were most relevant to the prediction [103].
  - **Deep Taylor Decomposition**: In relation to LRP, the Deep Taylor Decomposition breaks the output down into contributions from each input feature, visualizing the role of each pixel in a classification task [104].
- **Backpropagation Variants:**
  - **Guided Backpropagation and Deconvolution**: Both techniques focus on understanding CNN feature maps by propagating gradients back to the input, revealing influential patterns within an image. Deconvolution [105] uses filters in reverse, whereas Guided Backpropagation selectively visualizes positive gradients to highlight more interpretable image regions [106].
  - **PatternNet and PatternAttribution**: Proposed as improvements to gradient-based methods, these techniques attempt to accurately separate signal from noise, helping to correct gradient-based visualizations, especially in simple linear models [107].
- **Class Activation and Gradient-Based Localization Maps:**
  - **Class Activation Maps (CAM) and Grad-CAM**: CAM [108] and Grad-CAM generate class-specific localization maps by projecting weights from the output layer onto feature maps. Grad-CAM, a more flexible extension, works with a broader range of CNN architectures, offering coarse visualizations of regions critical to predictions [109].
  - **Grad-CAM++**: An enhancement of Grad-CAM, Grad-CAM++ improves the localization of multiple instances of an object in an image and supports multilabel classification tasks by using weighted gradients [110].

These methods collectively provide a range of tools for visualizing and interpreting the decision-making process of deep learning models, particularly in image

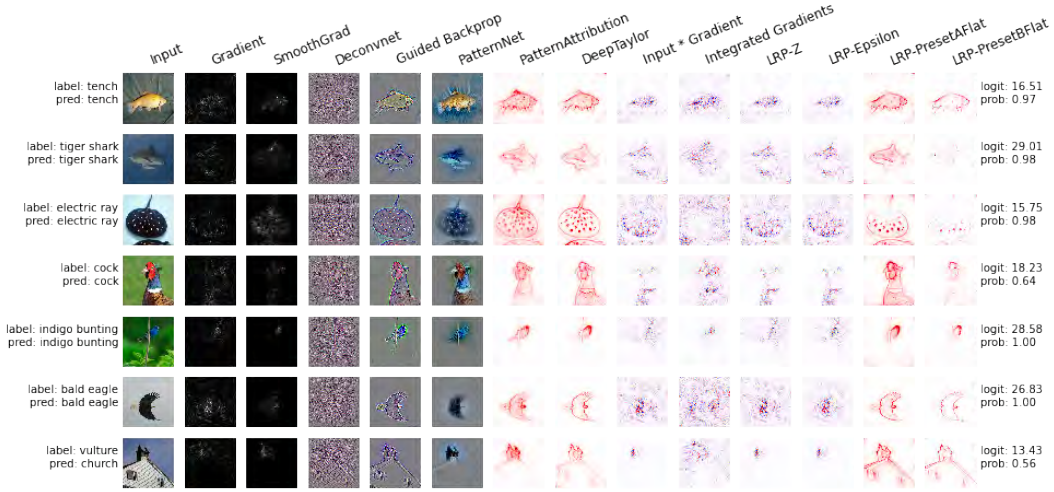

Figure 3.3: Comparison of Interpretability Methods for Deep Learning Models on ImageNet Sample Images, using the *innvestigate* package.

classification. Figure 3.3 provides a comparison of these interpretability methods, applied to sample images from the ImageNet dataset using the *innvestigate* [111] package. This visual comparison highlights how different techniques can reveal unique aspects of model behavior, offering insights into specific image regions that influence predictions.

Table 3.1: Overview of Interpretability Methods for Deep Learning Models. This table presents a curated list of prominent interpretability methods employed to explain the predictions of deep learning models. Each method is categorized based on its approach (Post Hoc), scope of interpretation (Local or Global), specificity to model architecture (Model Specific or Model Agnostic), applicable data types (Image, Text, Tabular), and the year of introduction. The table also highlights popular tools implementing these methods, with further implementation details and repository links provided in Appendix Table A.1.

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[99]	DeepExplain iNNvestigate tf-explain	Post Hoc	Local	Specific	Image	2013
[105]	DeepExplain iNNvestigate tf-explain	Post Hoc	Local	Specific	Image	2014

Continued on next page

Table 3.1 – continued from previous page

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[112]	iNNvestigate	Post Hoc	Local	Specific	Image	2014
[113]	Deep Visualization Toolbox	Post Hoc	Local	Specific	Image	2015
[114]	DeepExplain iNNvestigate The LRP Toolbox Skater	Post Hoc	Local	Specific	Image Text	2015
[108]	CAM	Post Hoc	Local	Specific	Image	2016
[115]	rationale	Post Hoc	Local	Specific	Text	2016
[109]	Grad-CAM tf-explain	Post Hoc	Local	Specific	Image	2017
[100]	DeepExplain iNNvestigate Integrated Gradients tf-explain alibi Skater	Post Hoc	Local	Specific	Image Text Tabular	2017
[116]	DeepExplain DeepLift iNNvestigate tf-explain Skater	Post Hoc	Local	Specific	Image	2017
[104]	iNNvestigate	Post Hoc	Local	Specific	Image	2017
[101]	iNNvestigate tf-explain	Post Hoc	Local	Specific	Image	2017
[117]	tcav	Post Hoc	Local	Specific	Image	2018
[110]	Grad-CAM++	Post Hoc	Local	Specific	Image	2018
[118]	RISE	Post Hoc	Local	Specific	Image	2018

Continued on next page

Table 3.1 – continued from previous page

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[107]	iNNvestigate	Post Hoc	Local	Specific	Image	2017

3.2.1.2 Interpretability Methods for Any Black-Box Model

Model-agnostic interpretability techniques have emerged as essential tools for understanding the decision-making processes of black-box models across diverse applications. Unlike methods tailored to specific architectures, such as deep learning, these approaches are applicable to any model or data type, making them highly versatile. Key techniques include perturbation-based methods, feature attribution, and counterfactual analysis, which provide insight into model behavior without altering its structure.

Table 3.2 categorizes these methods by interpretative approach, scope (local or global), model independence (agnostic), supported data types, and year of introduction. For practical implementation, the Appendix (Table A.1) provides a list of tools and repositories, enabling users to seamlessly apply these techniques in their workflows.

- **Local Interpretable Model-Agnostic Explanations (LIME)**: LIME [119] is a popular model-agnostic interpretability method that provides explanations for individual predictions by approximating the model locally around a specific instance. Garreau et al. [120] provided the first theoretical analysis of LIME, confirming its value but warning that poor parameter choices could lead to the overlooking of essential features. Figure 3.4 demonstrates LIME's application in explaining the classification rationale for an instance from the Quora Insincere Questions Dataset.
- **Partial Dependence Plots (PDPs)**: Proposed by Friedman [121], PDPs are a visualization tool that helps to interpret black-box models by showing the average impact of specific features on predictions. They plot how varying a feature or set of features affects predictions while marginalizing over other

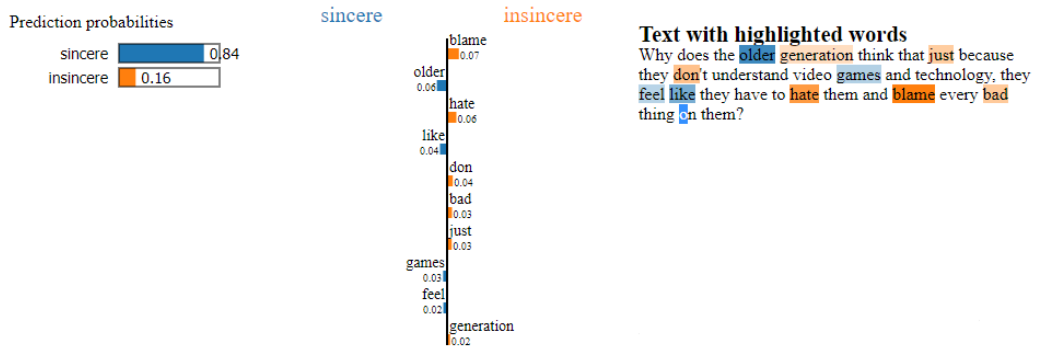

Figure 3.4: LIME explaining the classification rationale of an instance in the Quora Insincere Questions Dataset.

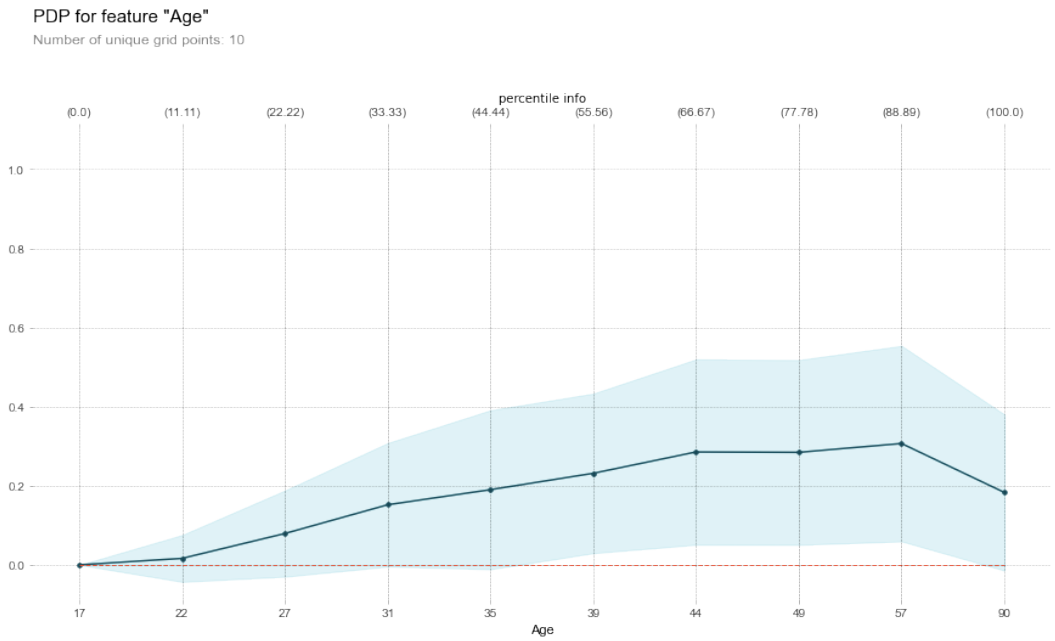

Figure 3.5: PDP for a Random Forest model, showing the relationship between age (feature) and income percentile (label) using the Census Income dataset (UCI Machine Learning Repository).

features. Although PDPs may not capture all feature interactions, they offer valuable insight, especially in cases with minimal interactions. Figure 3.5 shows a PDP for a Random Forest model, illustrating the relationship between age and income percentile in the Census Income dataset.

- **ICE (Individual Conditional Expectation) Plots**: Often related to PDPs, ICE plots visualize the effect of a feature on the predictions at the individual level, complementing PDPs by illustrating variations between in-

stances [122].

- **Shapley Additive Explanations (SHAP)**: A method that unifies existing feature attribution methods in a consistent framework, calculating feature importance values based on Shapley values from cooperative game theory [123]. SHAP provides explanations at both local and global scopes, although some variants (e.g., KernelSHAP) struggle with feature dependencies.
- **Anchors and Contrastive Explanations**: Unlike feature attribution methods, **Anchors** produce high-precision rules (if-then conditions) for local interpretation [124], while **Contrastive Explanations** identify both necessary and absent features crucial to a prediction, offering explanations that align closely with human reasoning [125].
- **Counterfactual Explanations**: Unlike internal model insights, counterfactual techniques suggest minimal changes in feature values to alter predictions, focusing on factors that could reverse the model's decisions [126].
- **Accumulated Local Effects (ALE)**: ALE plots adjust for feature dependencies in partial dependence analysis, providing an alternative to PDPs by calculating the impact of feature differences on predictions [127].

Table 3.2: Overview of Model-Agnostic Interpretability Methods for Black-Box Models. This table presents a curated list of interpretability methods designed to explain the behavior of any black-box model, irrespective of architecture or data type. Each method is categorized by its approach (Post Hoc), scope of interpretation (Local or Global), and model independence (Model Agnostic). It also outlines applicable data types (e.g., Image, Text, Tabular) and the year of introduction. Additionally, tools implementing these methods are identified, with further implementation details and repository links available in Appendix Table A.1.

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[121]	PDPbox InterpretML Skater	Post Hoc	Global	Agnostic	Tabular	2001
[128]	Eli5	Post Hoc	Global	Agnostic	Tabular	2010

Continued on next page

Table 3.2 – continued from previous page

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[122]	PyCEbox	Post Hoc	Local & Global	Agnostic	Tabular	2015
[119]	lime Eli5 InterpretML AIX360 Skater	Post Hoc	Local	Agnostic	Image Text Tabular	2016
[127]	alibi	Post Hoc	Global	Agnostic	Tabular	2016
[123]	shap alibi AIX360 InterpretML	Post Hoc	Local & Global	Agnostic	Image Text Tabular	2017
[126]	alibi	Post Hoc	Local	Agnostic	Tabular Image	2017
[124]	alibi Anchor	Post Hoc	Local	Agnostic	Image Text Tabular	2018
[129]	pyBreakDown	Post Hoc	Local & Global	Agnostic	Tabular	2018
[130]	L2X	Post Hoc	Local	Agnostic	Image Text Tabular	2018
[125]	alibi AIX360	Post Hoc	Local	Agnostic	Tabular Image	2018
[131]	DLIME	Post Hoc	Local	Agnostic	Image Text Tabular	2019
[132]	AIX360	Post Hoc	Local	Agnostic	Tabular	2019

Continued on next page

Table 3.2 – continued from previous page

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[133]	AIX360	Post Hoc	Local	Agnostic	Tabular Image	2019
[134]	alibi	Post Hoc	Local	Agnostic	Tabular Image	2019

3.2.2 Interpretability Methods to Create White-Box Models

This category includes methodologies for creating models that are inherently interpretable and comprehensible by humans, commonly referred to as intrinsic, transparent, or white-box models. Examples include linear models, decision trees, and rule-based systems, as well as advanced models that offer similar transparency and hold significant promise in the interpretability domain. This discussion focuses on more sophisticated models, as basic interpretable models such as linear and decision tree models have been extensively covered in prior research.

Table 3.3 provides a summary of key techniques for building white-box models, detailing their scope (local or global), the specificity of the model architecture (model specific or model agnostic), the supported data types and the year of introduction. The table also identifies the tools that implement these methods and serves as a guide for researchers and practitioners interested in using these techniques. For further details on these tools and their implementation, refer to the comprehensive repository links listed in the Appendix Table A.1.

- **Supersparse Linear Integer Models (SLIM)**: Ustun and Rudin [135] introduced SLIM, a predictive system restricted to basic arithmetic operations such as addition, subtraction, and multiplication on input features, ensuring high interpretability.
- **Generalized Additive Models with Pairwise Interactions (GA^{2}Ms)**: Microsoft's work [136], based on [137], presented GA^2 Ms - generalized additive models enhanced with pairwise interactions. These models lever-

age modern techniques such as bagging and boosting while addressing co-linearity with a round-robin boosting approach. By automatically identifying and including pairwise interactions, GA^{2}Ms combine high accuracy with natural interpretability, making them suitable for critical applications such as healthcare.

- **Boolean Rule Column Generation**: Dash et al. [138] proposed a method that employs Boolean rules in disjunctive or conjunctive normal forms (DNF/CNF) for interpretability. Simplified rules with minimal clauses improve human comprehension. To address computational challenges with large datasets, a randomized column generation algorithm balances simplicity of rules and classification accuracy.
- **Generalized Linear Rule Models (GLRMs)**: Wei et al. [139] introduced GLRMs, which extend Generalized Linear Models (GLMs) [140] by incorporating rule-based features. This approach maintains interpretability while capturing non-linear dependencies and relationships, resulting in weighted rule combinations evolving during training.
- **TED Framework for Local Explanations**: Hind et al. [141] presented TED, a framework for domain-specific explanations. Instead of probing model reasoning, TED emulates the decision-making process of human experts, offering tailored, accessible explanations for specific domains.

Despite their promise, white-box models have seen limited advancements due to the growing complexity of applications and trade-offs in predictive power. Their limitations are particularly evident in fields like computer vision and natural language processing, where deep learning dominates. Additionally, the rise of multi-task and transfer learning diminishes the relevance of white-box models, which are primarily designed for single-task performance.

3.2.3 Interpretability Methods to Enhance Fairness in Machine Learning Models

Machine learning systems are increasingly integrated into real-world applications, making fairness and discrimination mitigation essential. The subfield of ML fair-

Table 3.3: Overview of Interpretability Methods for Constructing White-Box Models. This table presents key methodologies for creating interpretable models designed to be intrinsically transparent and comprehensible by humans. The methods are categorized by their scope (Local or Global), specificity to model architecture (Model Specific or Model Agnostic), and supported data types (e.g., Tabular). Tools implementing these methods are also highlighted, offering a practical perspective on their application. For implementation resources and repository links, see Appendix Table A.1.

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[136]	InterpretML	White Box	Global	Specific	Tabular	2015
[135]	Slim	White Box	Global	Specific	Tabular	2016
[138]	AIX360	White Box	Global	Specific	Tabular	2018
[141]	AIX360	White Box	Local	Specific	Tabular	2019
[139]	AIX360	White Box	Global	Specific	Tabular	2019

ness focuses on evaluating algorithms for social and ethical impacts, emphasizing impartiality and mitigating bias. Research in this area has advanced significantly, introducing methods to ensure fairness through pre-processing, algorithmic adjustments, and post hoc corrections.

This section categorizes these fairness methods into themes, highlighting key approaches and tools that have been developed to address bias in machine learning models. Table 3.4 provides a comprehensive summary of these methodologies, detailing their scope (Local or Global), model dependence (Model Specific or Model Agnostic), supported data types and year of introduction. For readers interested in exploring practical implementations, the Appendix Table A.1 lists the tools and corresponding repository links, offering direct access to resources for further study and application.

- **Fairness Testing and Detection**
  - **Disparate Impact Testing**: Evaluates fairness by comparing predictions across demographic groups. Although efficient, it lacks information on the causes of bias and may not detect local discrimination in complex models [142]. An example of parity testing, showing disparities in prediction metrics between demographic groups using the Ae-

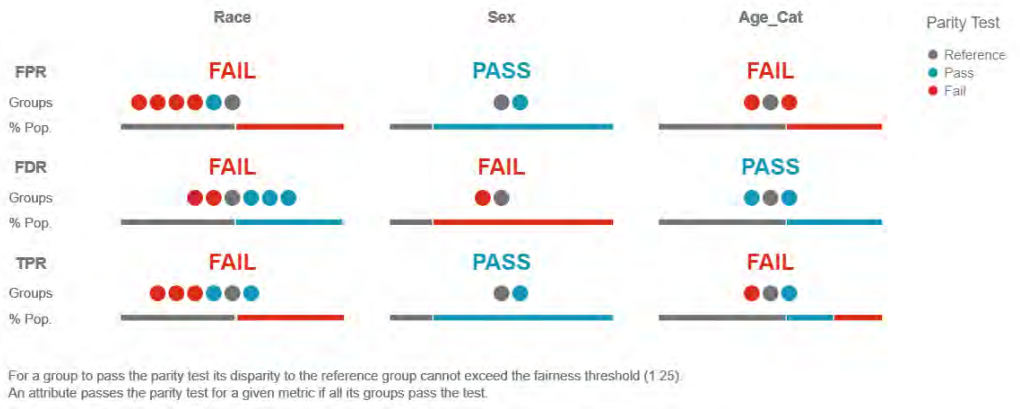

Figure 3.6: Parity testing using the Aequitas library on the ProPublica COMPAS dataset, demonstrating disparities across demographic groups.

quitas library, is shown in Figure 3.6.

  - **Counterfactual Fairness**: Defines fairness as the consistency of decisions across actual and hypothetical demographic groups, employing causal analysis to address biases [143].
  - **Bias in Word Embeddings**: Bolukbasi et al. [144] revealed gender biases in word embeddings and proposed methods to mitigate these biases in textual data.

- **Preprocessing Techniques for Fairness**
  - **Feature Suppression and Data Massaging**: Kamiran et al. [145] proposed methods to reduce biases in training data by removing sensitive attributes or adjusting instance labels. Similarly, Calders and Verwer [146] introduced three approaches to discrimination-free classification: unawareness (removing sensitive features), massaging (relabeling instances) and preferential sampling (adjusting sampling proportions to balance groups).
  - **Reweighing Instances**: Adjusts the importance of the instance according to demographic proportions, avoiding direct label alterations [147].
  - **Optimized Data Representation**: Calmon et al. [148] introduced a convex optimization approach to balance utility, fairness, and minimal data distortion.

- **Algorithmic Adjustments for Fairness**
  - **Adversarial Debiasing**: Prevents demographic bias by training models to minimize prediction dependency on sensitive attributes [149].
  - **Discrimination-Aware Classifiers**: Reject Option-based Classification and Discrimination-Aware Ensembles use decision theory to reduce bias without data modifications [150].
  - **Meritocratic Fairness**: Frameworks ensuring fairness in sequential decision making through reward optimization and prevention of preferential treatment [151].
- **Fairness in Dynamic and Complex Environments**
  - **Temporal Modeling**: The authors emphasize the importance of evaluating fairness over time, demonstrating the long-term effects of seemingly fair rules [152].
  - **Strategic Manipulation Models**: Studies on adaptive behaviors revealed disproportionate burdens on disadvantaged groups, highlighting the need for fairness-based decision boundaries [153, 154].
- **Fairness in Resource Allocation**
  - **Optimizing Fair Allocations**: Algorithms for resource distribution demonstrated in predictive police applications [155].
  - **Addressing Feedback Loops**: Methods to mitigate bias amplification in predictive policing [156]. Figure 3.7 illustrates a comparison of the levels of racial bias (bias disparity) between different demographic groups.
- **Methodological Enhancements**
  - **Fair Classification Frameworks**: A two-part framework combining similarity metrics with statistical parity for fair classification [157].
  - **Fairness and Calibration**: Explores the challenges of achieving fairness alongside calibrated probability scores, proposing post-processing solutions [158].

Fairness in machine learning is an evolving domain that has introduced meth-

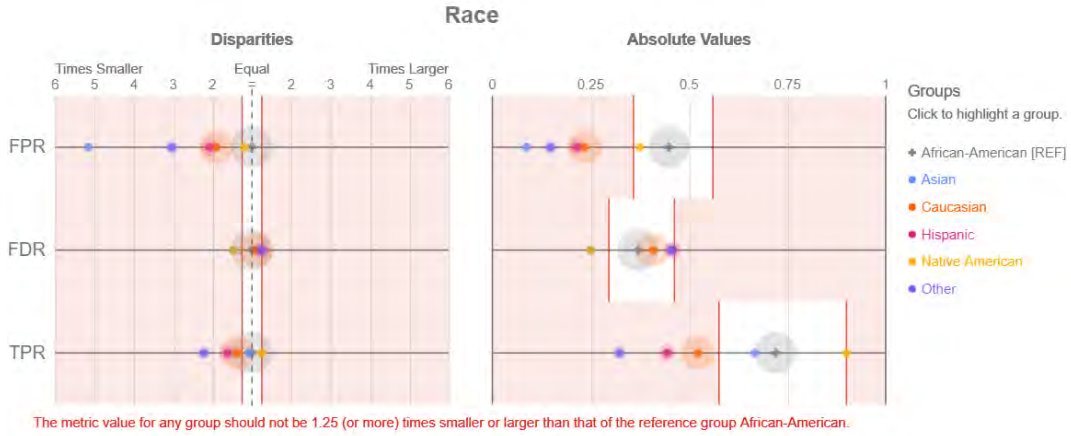

Figure 3.7: Comparison of racial bias levels (bias disparity) among different demographic groups in a sample population.

ods that target biases in data, algorithms, and post hoc analyses. However, most methods emphasize group-level fairness, overlooking individual-level factors. Additionally, fairness in non-tabular domains like images and text remains an underexplored area, representing opportunities for future research.

Table 3.4: Overview of Fairness Methods for Mitigating Bias and Discrimination in Machine Learning Models. This table presents a curated list of interpretability and fairness techniques aimed at restricting discrimination and promoting fairness in machine learning models. Each method is categorized by its scope (Local or Global), model dependence (Model Specific or Model Agnostic), and applicable data types (e.g., Tabular, Text). The table also identifies tools implementing these methods, providing a practical perspective on their application. Detailed repository links for these tools are available in Appendix Table A.1.

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[147]	AIF360	Fairness	Local	Agnostic	Tabular	2009
[159]	themis-ml	Fairness	Local	Agnostic	Tabular	2009
[146]	fairness-comparison	Fairness	Global	Specific	Tabular	2010
[145]	AIF360	Fairness	Global	Agnostic	Tabular	2012
[157]	fairness	Fairness	Local	Agnostic	Tabular	2012

Continued on next page

Table 3.4 – continued from previous page

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[150]	AIF360 themis-ml	Fairness	Global	Agnostic	Tabular	2012
[160]	AIF360	Fairness	Global	Agnostic	Tabular	2012
[161]	AIF360	Fairness	Local & Global	Agnostic	Tabular	2013
[142]	Aequitas AIF360 themis-ml	Fairness	Global	Agnostic	Tabular	2015
[162]	equalized_odds_and_calibration fairlearn AIF360	Fairness	Global	Agnostic	Tabular	2016
[144]	debiaswe	Fairness	Local	Specific	Text	2016
[151]	FairMachineLearning	Fairness	Local	Specific	Tabular	2016
[163]	fair-classification	Fairness	Global	Agnostic	Tabular	2017
[143]	fairness-in-ml	Fairness	Local	Agnostic	Tabular	2017
[164]	fair-classification	Fairness	Global	Agnostic	Tabular	2017
[148]	AIF360	Fairness	Global	Agnostic	Tabular	2017
[165]	fair-classification	Fairness	Global	Agnostic	Tabular	2017
[158]	equalized_odds_and_calibration AIF360	Fairness	Global	Agnostic	Tabular	2017
[166]	fairlearn	Fairness	Global	Agnostic	Tabular	2018
[149]	AIF360	Fairness	Local & Global	Agnostic	Tabular	2018
[167]	AIF360 GerryFair	Fairness	Global	Agnostic	Tabular	2018
[152]	ML-fairness-gym	Fairness	Global	Agnostic	Tabular	2018

Continued on next page

Table 3.4 – continued from previous page

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[156]	ML-fairness-gym	Fairness	Global	Agnostic	Tabular	2018
[168]	procedurally_fair_learning	Fairness	Global	Agnostic	Tabular	2018
[169]	AIF360	Fairness	Global	Agnostic	Tabular	2019
[154]	ML-fairness-gym	Fairness	Local	Specific	Tabular	2019
[153]	ML-fairness-gym	Fairness	Local	Specific	Tabular	2019
[155]	ML-fairness-gym	Fairness	Global	Specific	Tabular	2019

3.2.4 Interpretability Methods to Analyze the Sensitivity of Machine Learning Model Predictions

This category focuses on interpretability techniques designed to evaluate and scrutinize machine learning models to ensure that their predictions are reliable and trustworthy. These techniques employ sensitivity analysis to examine the stability of the model's learned functions and to evaluate how responsive the predictions are to small, deliberate modifications in the input data. Sensitivity analysis can be used as either a global or local interpretability approach, depending on whether the analysis pertains to the model's behavior for a single instance or across the entire dataset. This category explores both traditional methods and adversarial example-based sensitivity approaches, with summaries of each provided in Tables 3.5 and 3.6, respectively.

3.2.4.1 Traditional Sensitivity Analysis Methods

Traditional sensitivity analysis methods aim to quantify the influence of input variables on the output of a model using metrics called sensitivity indices. These indices are categorized as first-order indices, which measure the direct contribution of an individual input to output variance, and higher-order indices, which quantify the impact of interactions among two or more inputs. In addition, total effect indices combine the contributions of first-order and higher-order effects to account for all

influences on output variance.

Table 3.5 provides an overview of traditional sensitivity analysis techniques, highlighting key methods, their scope (Local or Global), model independence (Model Specific or Model Agnostic), supported data types, and year of introduction. The table also identifies tools, such as SALib, that implement these methods. For practical resources, including repository links for implementation, see Appendix Table A.1.

Sobol's work formalized sensitivity analysis based on variance decomposition, introducing techniques to approximate first-order and higher-order indices through Monte Carlo methods [170]. Based on this, Saltelli et al. improved the Sobol method with more efficient sampling strategies, allowing the calculation of first-order, higher-order, and total-effect indices [171, 172].

The Fourier Amplitude Sensitivity Test (FAST), developed by Cukier et al., advanced sensitivity analysis by transforming multi-dimensional integrals into one-dimensional ones via Fourier transformations [173]. This innovation improved the computation of the Sobol indices. Saltelli et al. further refined FAST to estimate total-effect indices [174], while Tarantola et al. extended Random Balance Designs (RBD), originally used in regression by Satterthwait, for application in non-linear and non-additive models by integrating RBD with FAST (RBD-FAST) [175]. Plischke introduced sampling improvements to increase the computational efficiency of RBD-FAST [176], and Tissot et al. added bias correction methods for enhanced estimation accuracy [177].

Another global sensitivity analysis approach is the Morris method, commonly referred to as the One-at-a-Time (OAT) method [178]. This method categorizes input variables into three groups: those with negligible effects, those with significant linear effects but no interactions, and those with non-linear or interactive effects. It involves discretizing the input space and iterating one variable at a time.

The Morris method, while complete, is very costly and, as a result, in some cases, fractional factorial designs, as described in [179], need to be formulated and used in practice to perform sensitivity analysis more efficiently. By devising a more

effective sampling strategy as well as other improvements, Campolongo et al. [180] proposed an improved version of Morris' method.

In some cases, variance is not an ideal measure of sensitivity. Alternative approaches, such as Borgonovo's density-based sensitivity indices, measure the divergence between unconditional and conditional output distributions without relying on variance calculations [181]. Plischke et al. expanded on Borgonovo's work by developing new estimators for these indices that are independent of the sampling method used [182].

Derivative-Based Global Sensitivity Measures (DGSM), introduced by Kucherenko and Song, estimate sensitivity by averaging local derivatives [183]. These methods, which use Monte Carlo or quasi-Monte Carlo techniques, are simpler to implement than Sobol indices and can be considered an extension of the Morris method.

3.2.4.2 Adversarial Example-based Sensitivity Analysis Methods

Adversarial examples are data points modified with subtle perturbations designed to mislead machine learning models into incorrect predictions. Unlike counterfactual examples, which are used to explain model behavior, adversarial examples focus solely on misdirection. Methods for generating and analyzing adversarial examples have been developed across various domains that provide insight into model vulnerabilities and robustness. Table 3.6 provides a comprehensive summary of these techniques, categorized by their application domain, scope (local or global), and model dependence (Model Specific or Model Agnostic).

For readers interested in exploring practical implementations, the table also highlights tools such as CleverHans, Foolbox, and TextAttack, which are widely used for adversarial example-based sensitivity analysis. Details and repository links for these tools can be found in the Appendix Table A.1, providing resources for further exploration and experimentation.

- **Computer Vision Applications:**

  Szegedy et al. [184] first identified the vulnerability of deep neural networks to small input perturbations and introduced adversarial examples by formulat-

Table 3.5: Overview of Traditional Sensitivity Analysis Methods. This table presents a summary of traditional sensitivity analysis methods designed to evaluate the influence of input variables on a model's output. The methods are categorized by their scope (Local or Global), model independence (Model Specific or Model Agnostic), and supported data types (e.g., Tabular). Tools like SALib that implement these techniques are highlighted, offering practical insights into their application. For additional details and repository links, refer to Appendix Table A.1.

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[173]	SALib	Sensitivity	Global	Agnostic	Tabular	1973
[178]	SALib	Sensitivity	Global	Agnostic	Tabular	1991
[174]	SALib	Sensitivity	Global	Agnostic	Tabular	1999
[170]	SALib	Sensitivity	Global	Agnostic	Tabular	2001
[171]	SALib	Sensitivity	Global	Agnostic	Tabular	2002
[175]	SALib	Sensitivity	Global	Agnostic	Tabular	2006
[180]	SALib	Sensitivity	Global	Agnostic	Tabular	2007
[181]	SALib	Sensitivity	Global	Agnostic	Tabular	2007
[179]	SALib	Sensitivity	Global	Agnostic	Tabular	2008
[176]	SALib	Sensitivity	Global	Agnostic	Tabular	2010
[172]	SALib	Sensitivity	Global	Agnostic	Tabular	2010
[177]	SALib	Sensitivity	Global	Agnostic	Tabular	2012
[182]	SALib	Sensitivity	Global	Agnostic	Tabular	2013
[183]	SALib	Sensitivity	Global	Agnostic	Tabular	2016

ing the problem as an L_2 norm optimization task. Goodfellow et al. [185] proposed the Fast Gradient Sign Method (FGSM), which uses L_∞ norm to generate adversarial examples, attributing this vulnerability to neural networks' linear properties (Figure 3.8). Moosavi-Dezfooli et al. [186] introduced DeepFool, a method for crafting minimal perturbation adversaries through iterative classifier linearization. They later proposed universal perturbations [187], which apply broadly across diverse inputs. Carlini and Wagner [188] developed effective adversarial attacks based on L_0, L_2, and L_∞ norms by optimizing the margin loss, improving on earlier methods. Xiao et al. [189] introduced spatial transformations (e.g., translation, rotation) to craft imperceptible yet effective perturbations. Su et al. [190] demonstrated the "one-pixel

Figure 3.8: Fast Gradient Sign Method (FGSM) on the MNIST dataset. The first row contains unperturbed images, while subsequent rows show perturbed images using different ε values, resulting in misclassifications.

attack," which perturbs a single pixel under L_0 norm constraints. Dong et al. [191] introduced momentum-based optimization to stabilize and enhance iterative attack algorithms. Brendel et al. [192] proposed the Boundary Attack, a decision-based approach that learns the boundary between adversarial and benign examples to generate effective perturbations.

- **Natural Language Processing (NLP) Applications:**

  Jia and Liang [193] introduced ADDANY and ADDSENT attacks, which add misleading phrases to input data for adversarial examples in reading comprehension tasks. Samanta and Mehta [194] and Iyyer et al. [195] created adver-

sarial sentences using synonym substitutions and paraphrasing techniques. Ebrahimi et al. [196] proposed HOTFLIP to perturb individual characters, while Liang et al. [197] extended this to word- and phrase-level perturbations. Alzantot et al. [198] used genetic algorithms to perturb text embeddings, a method later refined by Wang et al. [199] for higher transferability and success rates. Cheng et al. [200] focused on sequence-to-sequence models, employing projected gradient techniques to address the challenges of discrete input domains. Garg and Ramakrishnan [201] and Li et al. [202] exploited the BERT masked language model to generate textual adversarial examples by substituting masked segments. Zang et al. [203] utilized sememe-based word substitutions, employing particle swarm optimization for efficient adversarial example generation.

- **Graph-Structured Data:**

  Adversarial attacks on graph data are less explored but show significant potential. Zügner et al. [204] proposed a greedy algorithm to attack node classification models by modifying graph edges and node features. Dai et al. [205] employed reinforcement learning to create graph adversarial examples by altering graph connections. Zügner et al. [206] further extended these ideas using meta-learning techniques to optimize graph attacks efficiently.

- **Broader Applications and Studies:**

  Huang et al. [116] demonstrated the susceptibility of reinforcement learning models to adversarial states created using FGSM, which significantly impacted learned policies. Cisse et al. [207] extended adversarial techniques to structured prediction tasks, including pose estimation and semantic segmentation. Rauber and Bethge [208] developed faster and more effective clipping methods for noise-based adversarial examples. Li et al. [209] introduced a probabilistic approach to adversarial example generation, fitting distributions over adversarial regions to attack models without requiring knowledge of their internals.

This review demonstrates the extensive vulnerabilities of machine learning

models to adversarial attacks across domains such as computer vision, natural language processing, and graph-structured data. While vision and NLP applications dominate the research landscape, adversarial techniques for tabular data and other less-explored domains present opportunities for future research.

Table 3.6: Overview of Adversarial Example-based Sensitivity Analysis Methods. This table presents a comprehensive list of adversarial example-based sensitivity analysis techniques, detailing their scope (Local or Global), model dependence (Model Specific or Model Agnostic), and supported data types (e.g., Image, Text, Graph). Tools such as CleverHans, Foolbox, and TextAttack, which implement these methods, are also highlighted. Additional implementation details and repository links are provided in Appendix Table A.1.

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[184]	cleverhans foolbox	Sensitivity	Local & Global	Agnostic	Image	2013
[185]	cleverhans foolbox	Sensitivity	Local & Global	Agnostic	Image	2014
[210]	cleverhans foolbox	Sensitivity	Local & Global	Agnostic	Image	2016
[186]	cleverhans foolbox	Sensitivity	Local & Global	Agnostic	Image	2016
[211]	cleverhans foolbox	Sensitivity	Local & Global	Agnostic	Image	2016
[212]	transferability-advdnn-pub	Sensitivity	Local & Global	Agnostic	Image	2016
[213]	accessorize-to-a-crime	Sensitivity	Local & Global	Agnostic	Image	2016
[214]	adversarial_text adversarial_training adversarial_training_methods	Sensitivity	Local & Global	Specific	Text	2016
[215]	foolbox	Sensitivity	Local & Global	Agnostic	Image	2016

Continued on next page

Table 3.6 – continued from previous page

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[188]	cleverhans nn_robust_attacks	Sensitivity	Local & Global	Agnostic	Image	2017
[216]	cleverhans	Sensitivity	Local & Global	Agnostic	Image	2017
[217]	influence-release	Sensitivity	Local & Global	Agnostic	Image	2017
[193]	adversarial-squad	Sensitivity	Local & Global	Specific	Text	2017
[218]	ZOO-Attack	Sensitivity	Local & Global	Agnostic	Image	2017
[192]	foolbox boundary-attack	Sensitivity	Local & Global	Agnostic	Image	2017
[219]	nn_robust_attacks	Sensitivity	Local & Global	Agnostic	Image	2017
[196]	WordAdver	Sensitivity	Local & Global	Agnostic	Text	2017
[220]	cleverhans	Sensitivity	Local & Global	Agnostic	Image	2017
[187]	universal	Sensitivity	Local & Global	Agnostic	Image	2017
[191]	cleverhans	Sensitivity	Local & Global	Agnostic	Image	2018
[204]	nettack	Sensitivity	Local & Global	Specific	Graph	2018
[198]	nlp_adversarial_examples	Sensitivity	Local & Global	Agnostic	Text	2018

Continued on next page

Table 3.6 – continued from previous page

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[195]	scpn	Sensitivity	Local & Global	Agnostic	Text	2018
[189]	stAdv	Sensitivity	Local & Global	Agnostic	Image	2018
[205]	Graph_adversarial_attack	Sensitivity	Local & Global	Specific	Graph	2018
[221]	foolbox AnalysisBySynthesis	Sensitivity	Local & Global	Agnostic	Image	2018
[222]	TextAttack	Sensitivity	Local & Global	Agnostic	Text	2018
[223]	UAN	Sensitivity	Local & Global	Agnostic	Image	2018
[224]	TextAttack	Sensitivity	Local & Global	Agnostic	Text	2018
[225]	TextAttack	Sensitivity	Local & Global	Agnostic	Text	2018
[226]	TextAttack	Sensitivity	Local & Global	Agnostic	Text	2018
[190]	one-pixel-attack-keras	Sensitivity	Local & Global	Agnostic	Image	2019
[227]	foolbox	Sensitivity	Local & Global	Agnostic	Image	2019
[199]	TextAttack	Sensitivity	Local & Global	Agnostic	Text	2019
[208]	foolbox	Sensitivity	Local & Global	Agnostic	Image	2019

Continued on next page

Table 3.6 – continued from previous page

Ref	Tool	Category	Local vs. Global	Model Specific vs. Model Agnostic	Data Type	Year
[206]	gnn-meta-attack	Sensitivity	Local & Global	Specific	Graph	2019
[228]	HSJA	Sensitivity	Local & Global	Agnostic	Image	2019
[229]	TextAttack	Sensitivity	Local & Global	Agnostic	Text	2019
[209]	Nattack	Sensitivity	Local & Global	Agnostic	Image	2019
[230]	TextAttack TextFooler	Sensitivity	Local & Global	Specific	Text	2019
[200]	TextAttack	Sensitivity	Local & Global	Specific	Text	2020
[203]	TextAttack	Sensitivity	Local & Global	Agnostic	Text	2020
[202]	TextAttack	Sensitivity	Local & Global	Specific	Text	2020
[231]	TextAttack	Sensitivity	Local & Global	Agnostic	Text	2020
[201]	TextAttack	Sensitivity	Local & Global	Specific	Text	2020

3.3 Discussion

The primary contribution of this chapter is the establishment of a comprehensive taxonomy for machine learning interpretability methods. This taxonomy categorizes techniques into four main domains: methods for explaining complex black-box models, approaches for developing inherently interpretable white-box models, strategies to promote fairness and mitigate bias, and techniques for assess-

ing sensitivity in model predictions. By organizing interpretability methods across these dimensions, the taxonomy provides a structured framework to understand the strengths, limitations, and applicability of various techniques.

3.3.1 Advancements in Black-Box Model Explanation

A significant focus of interpretability research has been on explaining predictions from deep learning and other black-box models, particularly in fields like computer vision and natural language processing. Many methods revolve around visualizing decision-making processes, with saliency-based techniques such as Grad-CAM [109], an extension of CAM [108], which offers insight into neural network activations. Other techniques, like deconvolutional networks [105], further enhance understanding by reconstructing input features that drive predictions. For broader black-box models, LIME [119] and SHAP [123] remain dominant due to their model-agnostic flexibility and robust theoretical foundations. Despite their widespread adoption, simpler tools, such as partial dependency plots (PDP) [121], continue to play a valuable role, especially in structured data tasks.

3.3.2 Challenges in Building Transparent White-Box Models

White-box models inherently offer interpretability, but often struggle to match the predictive performance of complex black-box models in tasks such as image classification or natural language understanding. Techniques such as generalized additive models with pairwise interactions (GA2Ms) [136], based on [232], have achieved state-of-the-art performance in healthcare and other domains. However, their limited applicability to multitask and cross-domain problems reduces their appeal in modern machine learning pipelines, which increasingly demand versatile models capable of handling diverse and unstructured data.

3.3.3 Fairness and Bias Mitigation

Fairness and bias mitigation have gained prominence as machine learning systems are deployed in sensitive applications such as healthcare, finance, and the judicial system. Methods such as the Hardt et al. framework for reducing discrimination [162] have laid the foundational groundwork in this area. However, the major-

ity of fairness research remains focused on tabular datasets, with relatively limited progress in non-tabular domains such as images, text, and graphs. The integration of fairness into mainstream machine learning workflows remains a challenge, compounded by the need for domain-specific fairness metrics and techniques.

3.3.4 Sensitivity Analysis and Adversarial Robustness

Sensitivity analysis, particularly in the context of adversarial robustness, has seen rapid growth. The discovery of adversarial examples by Szegedy et al. [184] revealed the susceptibility of deep learning models to imperceptible input perturbations, causing a surge in research on adversarial defenses and robustness. These techniques range from general-purpose methods to task-specific adaptations for structured and unstructured data. The intersection of interpretability and adversarial analysis remains an emerging area with the potential to bridge the gap between understanding model vulnerabilities and mitigating their impact.

3.3.5 Barriers to Adoption

Despite significant advancements in explainable artificial intelligence (XAI), several challenges hinder its seamless integration into real-world machine learning workflows. Inconsistencies in terminology and a lack of standardization often complicate the adoption of interpretability techniques across different domains. Furthermore, while much of the research on interpretability provides valuable theoretical insights, its practical utility in improving real-world systems remains limited.

Another barrier lies in the usability gap: Many interpretability methods are designed with researchers in mind and may not align with the needs of practitioners in applied fields. Current techniques often fail to account for the specific requirements of different industries, such as regulatory compliance in healthcare or explainability standards in finance. Additionally, interpretability methods tend to focus on tabular or single-modality data, with limited exploration in more complex settings, such as multi-modal or streaming data.

Although these challenges underscore the need for further research and development, they also highlight the transformative potential of interpretability tech-

niques to enhance machine learning workflows. By addressing these barriers, the wider adoption of XAI can foster greater transparency, trust, and accountability in critical applications.

3.4 Summary and Next Steps

This chapter established a comprehensive taxonomy of machine learning interpretability methods, providing a structured framework for understanding and comparing different approaches. This taxonomy revealed key insights into the current state of interpretability research, as well as notable gaps, particularly in the domain of unsupervised learning.

A notable observation from this review is the strong emphasis on interpretability techniques for supervised learning tasks, especially in fields like computer vision and natural language processing. For example, methods such as Grad-CAM [109] and deconvolutional networks [105] excel at generating saliency maps to explain image classification models. Similarly, techniques such as LIME [119] and SHAP [123] are widely adopted for feature importance and interaction analysis in black-box models. However, the review highlights a pronounced gap in interpretability for unsupervised learning, where the lack of labeled data introduces unique challenges.

A particularly critical gap exists in interpretable methods for unsupervised outlier detection, a field essential for high-stakes applications such as fraud detection, anomaly identification, and network security. Current approaches often rely on opaque black-box algorithms that lack transparency, making it difficult to trust or validate their outputs. This lack of interpretability poses significant barriers to adoption in critical domains where trust, accountability, and regulatory compliance are paramount. Without clear explanations, these systems can hinder actionable insights and decision-making, especially in scenarios where outlier detection has significant consequences.

To address these challenges, the next chapter introduces an innovative method for unsupervised outlier detection. Building on the insights from this taxonomy, the

proposed method integrates concepts from interpretability frameworks into a novel approach tailored to unsupervised learning. This method focuses on addressing the dual challenges of interpretability and the absence of labeled data, providing both transparency and effectiveness in detecting anomalies. The next chapter explores the theoretical foundations, design principles, and practical evaluation of this method, demonstrating its ability to provide clear, actionable insights across diverse datasets.

Chapter 4

An Interpretable Method Combining Normalizing Flows with Decision Trees

Unsupervised outlier detection methods often lack transparency, creating significant challenges for their adoption in critical domains such as healthcare and security, where trust and explainability are essential. In these high-stakes applications, it is important not only to accurately identify anomalies, but also to provide clear explanations for model decisions. However, most existing methods operate as "black boxes," offering little insight into their reasoning processes [29]. This lack of interpretability complicates deployment, particularly in scenarios that require both local interpretability (why a specific point is flagged as an outlier) and global interpretability (understanding broader decision patterns).

As highlighted in Chapter 3, interpretability methods for unsupervised learning, particularly in the context of outlier detection, remain underdeveloped compared to supervised learning. Supervised model techniques, such as post hoc analyses, are often ill-suited for unsupervised settings, where the absence of labeled data introduces unique challenges [28].

To address these challenges, this chapter introduces a novel framework for interpretable unsupervised outlier detection that bridges the gap between performance and explainability. The proposed method integrates normalizing flows and deci-

sion trees, combining the generative modeling capabilities of deep learning with the transparency of interpretable models. Using normalizing flows, the method learns the underlying probability distribution of an unlabeled dataset and generates pseudolabels to classify data points based on their likelihood. These pseudolabels enable the application of supervised learning techniques, such as decision trees, within an unsupervised context. Decision trees offer clear decision rules and feature importance scores, providing actionable insights into the factors driving outlier detection.

The key contributions of this chapter are as follows.

- **Framework Innovation:** Introduction of a method that integrates normalizing flows and decision trees to address detection performance and interpretability in unsupervised outlier detection.
- **Comprehensive Evaluation:** Benchmarking against 23 state-of-the-art outlier detection algorithms on 17 datasets, demonstrating superior performance in metrics such as precision, recall, F1 score and Matthews correlation coefficient, with statistical significance validated using Friedman rankings and post hoc tests.
- **Scenario-Specific Insights:** Identification of conditions where the method excels and areas requiring refinement, based on performance variability across datasets.
- **Interpretability Demonstration:** Use of pseudolabels to enable supervised learning within an unsupervised framework, with clear explanations provided through visual tree diagrams, decision rules and feature importance analysis.

The remainder of this chapter is organized as follows. Section 4.1 reviews related work on interpretable unsupervised outlier detection and the use of normalizing flows, establishing the context and motivation of this study. Section 4.2 introduces the proposed framework, detailing the integration of normalizing flows with decision trees to address both performance and interpretability challenges. Section 4.3 describes the datasets, experimental design, and evaluation metrics used to validate the method. Section 4.4 presents comparative evaluations against state-of-the-art methods, together with an in-depth discussion of key findings. Finally,

Section 4.5 concludes the chapter by summarizing its contributions.

4.1 Related Work

This section reviews relevant studies in interpretable unsupervised outlier detection and the application of normalizing flows for unsupervised anomaly detection. The focus is on bridging the gap between detection accuracy and interpretability while addressing challenges in various domains. Interpretability in unsupervised outlier detection remains an underexplored but critical area, as emphasized by recent reviews [233], [234], [235]. This limitation challenges users in trusting and acting upon anomaly detection results.

4.1.1 Interpretable Unsupervised Outlier Detection

Autoencoders are a popular choice due to their reconstruction-based anomaly detection capabilities. Rajendran et al. [236], Gribbestad et al. [237], and Kieu et al. [238] applied these techniques in wireless signals, health monitoring, and time series data. Combining autoencoders with decision trees, Aguilar et al. [239] and Zenkl et al. [240] improved interpretability by leveraging the clear decision boundaries of tree models, particularly for categorical data.

Statistical and rule-based methods also have improved interpretability. Huang et al. [241] developed ISOD, using empirical cumulative distribution functions, while Barbado et al. [28] extracted interpretable rules from One-Class SVM models. These approaches provide transparency by linking anomalies to specific conditions or statistical thresholds.

Chawla et al. [242] used post hoc interpretability methods such as SHAP and LIME to reveal the contributions of features in the detection of network anomalies. Similarly, Carletti et al. [243] proposed DIFFI, enhancing the interpretability of the isolation forest by calculating the importance of cumulative features.

Hybrid models and novel approaches have emerged to balance detection performance and interpretability. An et al. [244] introduced AAU-Net, which combines sparse coding with neural networks for the detection of mechanical anomalies. Yang et al. [245] developed GRAM, employing gradient attention maps to explain graph

anomalies. Zhao et al. [246] presented FBLS, which integrates fuzzy rules with neural networks for enhanced interpretability.

In vehicle networks, Zhou et al. [247] combined Kalman Variational Autoencoders with subspace extraction layers to model normal behavior and distinguish anomalies using physical laws. For high-energy physics, Roche et al. [248] developed interpretable autoencoders based on decision trees for real-time anomaly detection.

The enhancements to Isolation Forest include SHAP-based modifications by Rachwal et al. [249], improving both interpretability and accuracy in fields such as fraud detection. Attention mechanisms have also been applied, such as Yang et al. [250] with MANet, which reduces noise that resembles outliers in seismic data while providing interpretability.

For electricity consumption data, Gao et al. [251] introduced GRU-MACGAN, generating synthetic anomalies and offering contrastive explanations to highlight feature contributions, addressing class imbalance and data quality issues.

These advancements, which span autoencoders, rule-based systems, fuzzy logic, post hoc explanations, and hybrid models, underscore ongoing efforts to make unsupervised outlier detection more interpretable and trustworthy for practical applications.

4.1.2 Normalizing Flows for Unsupervised Outlier Detection

Normalizing flows have gained prominence in unsupervised outlier detection due to their ability to transform simple base distributions into complex ones through invertible transformations, enabling precise likelihood estimation for unseen data. These methods have been applied across various domains, taking advantage of their flexibility and expressiveness.

In time series anomaly detection, notable methods include Masked Autoregressive Flows (MAF) and FFJORD [252], OmniAnomaly Integration of GRU and Planar Flows [253], and Graphical Normalizing Flows (GNF) for UAV flight logs [254]. Advanced techniques like DDANF [255] combine autoencoders with normalizing flows for multivariate anomaly detection.

For trajectory data, methods such as GRADINGS [256] estimate complex probability densities for segments and aggregate them into anomaly scores, effectively handling high-dimensional, variable-length data.

In image data, FastFlow [257] integrates 2D normalizing flows with feature extractors, and CFLOW-AD [258] uses conditional flows for the detection of patch-level anomalies in real time. Advanced applications include 3D facial scans for genetic syndromes [259] and medical imaging with AE-FLOW [260].

Video data applications include models such as STG-NF [261] for pose data and DA-Flow [262], which improves anomaly detection in skeletal data using attention modules. AGC-NF [263] introduces adaptive graph convolutions to improve anomaly detection in challenging scenarios.

For network traffic, GRAnD [264] combines VAEs and normalizing flows to robustly reconstruct inlier data while corrupting outliers. Similarly, FANFOLD [265] and other graph-based methods leverage flows for graph anomaly detection.

Methods addressing out-of-distribution (OOD) challenges include Kumar et al.'s integration of Maximum Mean Discrepancy (MMD) attention mechanisms [266], improving robustness by reducing overconfidence. Quantum Normalizing Flows [267] explore quantum architectures for bijective mappings, bypassing the need for labeled data.

Despite these advances, interpretability remains underexplored in normalizing flow-based models. Among 22 surveyed studies, only five explicitly address interpretability [253, 259, 260, 263, 268]. This gap underscores the need for approaches that not only detect anomalies, but also explain why data points are classified as outliers. Table 4.1 summarizes key studies in this area, highlighting their models, datasets, and interpretability features.

Table 4.1: Comparison of Studies on Unsupervised Outlier Detection Using Normalizing Flows, Including Datasets and Interpretability.

Reference	Year	#Models	# Datasets	Data Types	Interpretability
[252]	2019	1	2	Time Series	No
[253]	2019	5	3	Time Series	**Yes**
[256]	2020	2	6	Time Series	No
[269]	2021	2	1	Image	No
[257]	2021	20	3	Image	No
[266]	2021	5	6	Image	No
[258]	2022	9	2	Image/Video	No
[259]	2022	3	1	Image	**Yes**
[270]	2022	6	4	Image	No
[271]	2022	28	6	Video	No
[264]	2022	6	2	Tabular	No
[268]	2022	0	1	Tabular	**Yes**
[254]	2023	5	1	Time Series	No
[272]	2023	6	3	Image	No
[260]	2023	5	5	Image	**Yes**
[261]	2023	7	2	Video	No
[273]	2023	5	4	Image/Video	No
[255]	2024	12	3	Time Series	No
[262]	2024	26	5	Video	No
[263]	2024	10	3	Video	**Yes**
[265]	2024	9	15	Graph	No
[267]	2024	3	4	Tabular	No
Proposed	2024	23	17	Tabular/Image	**Yes**

4.2 Proposed Methodology

4.2.1 Overview

This study introduces a methodology that uses the robust distribution modeling capabilities of normalizing flows for unsupervised outlier detection. To enhance interpretability, decision trees are incorporated, providing clear, actionable insights by providing transparent explanations for outlier classification. This focus on interpretability addresses a significant challenge in unsupervised outlier detection.

Decision trees are particularly suitable for this purpose because of their inherent transparency, as 'white-box' models. In contrast to less interpretable models such as Support Vector Machines (SVM), decision trees create easily understandable if-then rules. Furthermore, they can capture non-linear dependencies between variables, a limitation in other interpretable models like logistic regression, which assumes linearity. Their tree-diagram representations add an additional layer of clarity, making them especially beneficial in applications where explainability is paramount. Additionally, while real-time prediction is not the primary focus of this work, decision trees enable much faster inference compared to models such as K-Nearest Neighbors (KNN), which is advantageous for time-sensitive decisions. This combination of interpretability, the ability to handle non-linear relationships, and computational efficiency makes decision trees highly effective in practical applications that require rapid and comprehensible model outputs.

The proposed methodology is summarized in Figure 4.1, which illustrates key steps such as data normalization, synthetic data generation, model training, evaluation, and feature importance analysis.

The pseudocode in Algorithm 2 provides a detailed blueprint of the approach. It outlines processes like training normalizing flows, generating synthetic data, pseudo-labeling outliers, training a decision tree, and performing feature importance analysis. This step-by-step representation facilitates the seamless implementation and reproducibility of the methodology.

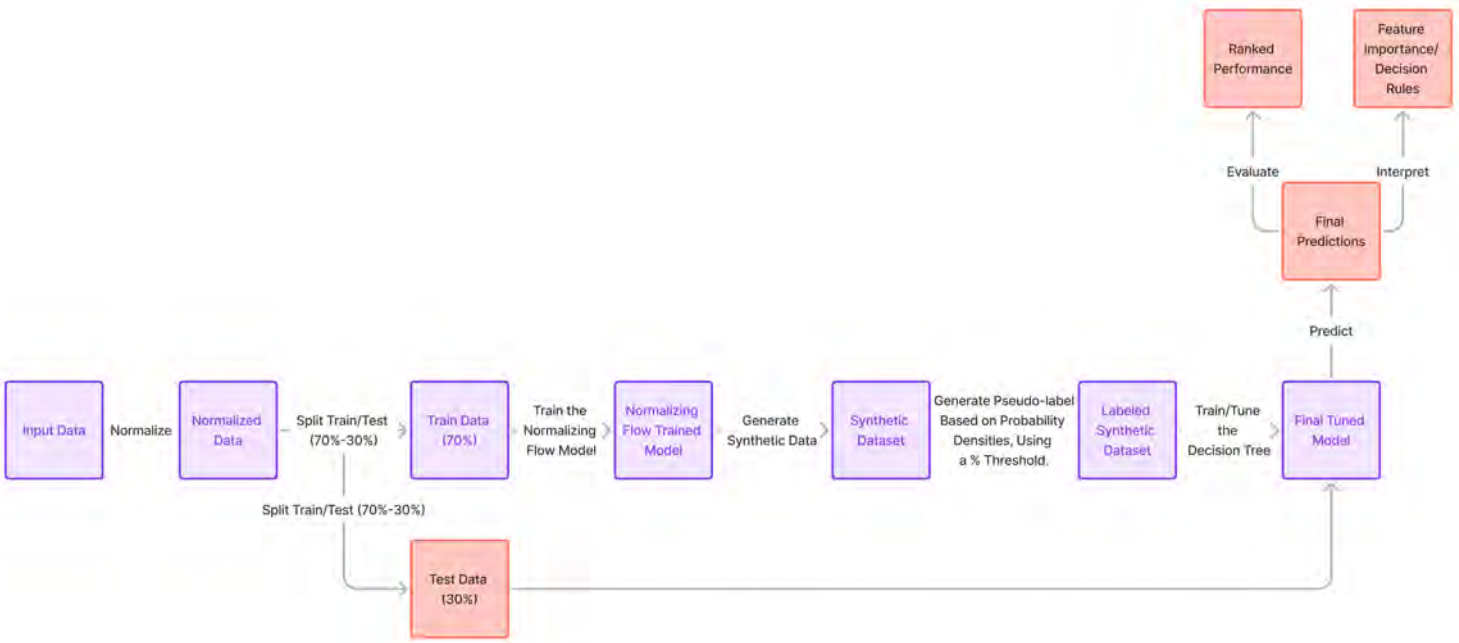

Figure 4.1: Flowchart of the Proposed Unsupervised Outlier Detection Method

4.2.2 Normalizing Flows for Probability Distribution Learning

Normalizing flows [274] are a class of advanced deep generative models designed to learn intricate data distributions through a series of invertible transformations. The central principle of normalizing flows involves transforming a simple base distribution, such as a standard multivariate normal distribution, into a complex target distribution that reflects the data's structure. This transformation is carried out through a sequence of bijective, differentiable mappings that enable an efficient calculation of the probability density function (PDF).

An illustrative example is presented in Figure 4.2.1, where a base distribution undergoes transformations, such as additive shifts and dimensional scaling, to evolve into a target distribution. This capability of mapping simple distributions to complex ones makes normalizing flows a powerful tool for unsupervised outlier detection in this methodology.

Algorithm 2 Algorithm for the Proposed Outlier Detection Method, including Normalizing Flow Training, Pseudo-Labeling, Decision Tree Training, and Interpretability Analysis

Step 1: Input
Dataset $\mathscr{D}$
Predefined probability threshold T
Step 2: Data Preparation
Normalize $\mathscr{D}$ to standardize input features.
Split $\mathscr{D}$ into training dataset $\mathscr{D}_{\text{train}}$ (70%) and testing dataset $\mathscr{D}_{\text{test}}$ (30%).
Step 3: Train Normalizing Flow Model
Train a Normalizing Flow model on $\mathscr{D}_{\text{train}}$.
Generate synthetic dataset $\mathscr{D}_{\text{synthetic}}$ using the trained Normalizing Flow model.
Step 4: Assign pseudolabels
Assign pseudolabels to $\mathscr{D}_{\text{synthetic}}$ based on the predefined probability threshold T.
Step 5: Train Decision Tree Model
Train and tune a decision tree model using $\mathscr{D}_{\text{synthetic}}$ with assigned pseudolabels.
Step 6: Test Model Predictions
Use the trained decision tree model to make predictions on $\mathscr{D}_{\text{test}}$.
Step 7: Evaluate Performance
Compare predictions with ground truth in $\mathscr{D}_{\text{test}}$ to evaluate the model's performance.
Step 8: Interpret Results
Compute feature importance from the decision tree model and interpret the results.
Step 9: Output
Evaluation metrics
Predictions for $\mathscr{D}_{\text{test}}$, based on the probability threshold T
Feature importance bar chart
Decision rules diagram

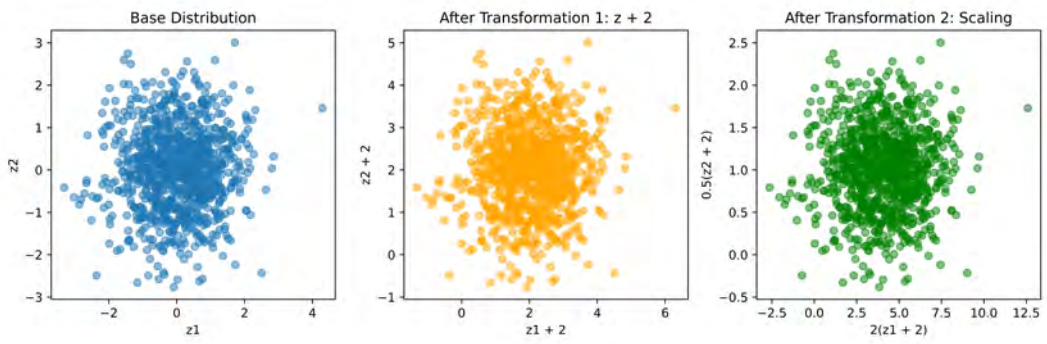

Figure 4.2: Normalizing Flows transforming a simple base distribution into a complex target distribution through a sequence of invertible transformations. The first transformation adds a constant vector to shift the distribution, and the second transformation scales the dimensions differently.

4.2.2.1 Mathematical Foundation

Given a dataset $\mathbf{x} \in \mathbb{R}^n$, the objective is to learn the probability distribution $p_{\mathbf{x}}(\mathbf{x})$. Normalizing flows accomplish this by applying a sequence of reversible transformations $f_1, f_2, \ldots, f_K$ to a simpler base distribution $p_z(\mathbf{z})$, where $\mathbf{z} \in \mathbb{R}^n$ originates from a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. This series of transformations maps the base distribution to the desired distribution as follows:

$$\mathbf{x} = f_K \circ f_{K-1} \circ \cdots \circ f_1(\mathbf{z}) \tag{4.1}$$

The probability density for $\mathbf{x}$ can then be calculated using the change-of-variables formula:

$$p_{\mathbf{x}}(\mathbf{x}) = p_z(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right|^{-1} \tag{4.2}$$

This can be further expressed using the chain rule as follows:

$$p_{\mathbf{x}}(\mathbf{x}) = p_z(\mathbf{z}) \prod_{k=1}^{K} \left| \det \left(\frac{\partial f_k}{\partial f_{k-1}} \right) \right|^{-1} \tag{4.3}$$

In this formulation, $\left| \det \left(\frac{\partial f_k}{\partial f_{k-1}} \right) \right|$ represents the absolute value of the determinant of the Jacobian matrix for the transformation f_k. The invertibility of the transformations ensures that the determinant can be computed efficiently.

In practical applications, computations are simplified by working with the logarithm of the probability density. This approach converts the product of the determinants into a sum, making the calculations more feasible. For multivariate data, the logarithmic form of the probability density is expressed as follows:

$$\log p_{\mathbf{x}}(\mathbf{x}) = \log p_z(f^{-1}(\mathbf{x})) + \log \left| \det \frac{df^{-1}(\mathbf{x})}{d\mathbf{x}} \right| \tag{4.4}$$

Here, $f^{-1}(\mathbf{x})$ denotes the inverse of the series of transformations applied to $\mathbf{x}$, and $\frac{df^{-1}(\mathbf{x})}{d\mathbf{x}}$ is the Jacobian of the inverse transformation.

4.2.2.2 Model Architecture

This section outlines the main components of the proposed architecture, which is built around Masked Autoregressive Flows (MAF) [275]. MAF was chosen for its ability to model complex data distributions while maintaining computational efficiency and interpretability.

1. Base Distribution

The architecture starts with a simple multivariate normal distribution $p_z(\mathbf{z})$, characterized by a zero-mean vector and a diagonal covariance matrix. This base distribution acts as the foundation for subsequent flow-based transformations:

$$p_z(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}) \tag{4.5}$$

This choice simplifies initial probability computations and ensures that transformations progressively refine the base distribution into the desired target distribution $p_{\mathbf{x}}(\mathbf{x})$.

2. Bijectors and Masked Autoregressive Flows

A key element of the architecture is the use of Masked Autoregressive Flows, where each layer applies an autoregressive transformation conditioned on preceding dimensions. This structure guarantees invertibility and efficient computation of the Jacobian determinant. The transformation applied by a single MAF layer is:

$$f_k(\mathbf{z}) = \mathbf{z} + m(\mathbf{z}) \odot s(\mathbf{z}) \tag{4.6}$$

Here, $m(\mathbf{z})$ and $s(\mathbf{z})$ are neural network-based functions with an autoregressive structure, and $\odot$ denotes the element-wise product (Hadamard). The Jacobian determinant for the transformation is computed as

$$\log \left| \det \frac{d f_k(\mathbf{z})}{d\mathbf{z}} \right| = \sum_{i=1}^{n} s_i(\mathbf{z}) \tag{4.7}$$

This design enables sequential transformations for each dimension, capturing dependencies within the data while maintaining invertibility. Figure 4.3 illustrates

the working of an MAF layer.

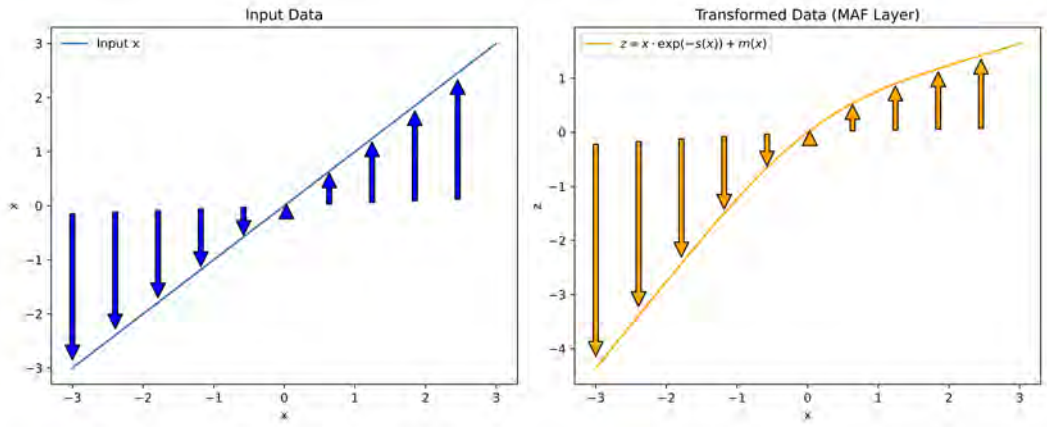

Figure 4.3: Illustration of a Masked Autoregressive Flow (MAF) layer. Dimensions are transformed sequentially based on preceding dimensions. The neural network functions are defined as $m(\mathbf{z}) = 0.5 * \mathbf{z}$ and $s(\mathbf{z}) = \log(1 + \exp(\mathbf{z}))$.

3. Permutation Bijectors

To enhance the flexibility of the flow model, permutation bijectors are introduced between MAF layers. These operations reorder dimensions, allowing the MAF layers to capture interactions between variables on different dimensions. A commonly used permutation is the reverse order:

$$\mathbf{z}' = \mathbf{P}\mathbf{z} \tag{4.8}$$

Here, $\mathbf{P}$ is a permutation matrix that reverses the order of the elements in $\mathbf{z}$.

4. Chain Bijector

The sequence of transformations, including MAF layers and permutation bijectors, is combined into a single chain. This chain forms a bijective mapping from the base distribution to the target distribution, transforming the simple Gaussian into a more complex data distribution:

$$f_{\text{chain}} = f_K \circ \mathbf{P}_{K-1} \circ f_{K-1} \circ \cdots \circ \mathbf{P}_1 \circ f_1 \tag{4.9}$$

5. Transformed Distribution

The chain bijector maps the base distribution to the transformed distribution $p_{\mathbf{x}}(\mathbf{x})$.

This transformation enables the computation of the log-probability for each data point, which is essential for training the model and performing outlier detection:

$$p_{\mathbf{x}}(\mathbf{x}) = p_z(f_{\text{chain}}^{-1}(\mathbf{x})) \left| \det \frac{df_{\text{chain}}(\mathbf{x})}{d\mathbf{x}} \right|^{-1} \tag{4.10}$$

4.2.2.3 Forward and Backward Pass

This section details the forward and backward passes in the normalizing flow architecture, focusing on the Masked Autoregressive Flow (MAF) layers and permutation operations within the chain bijector.

Forward Pass

For a given input vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, the forward pass proceeds as follows:

1. **First MAF Layer:**

   - For the first dimension:

$$z_1 = x_1 \cdot \exp(-s_1) + m_1 \tag{4.11}$$

   - For subsequent dimensions (conditioned on prior dimensions):

$$z_2 = (x_2 - m_2(x_1)) \cdot \exp(-s_2(x_1)) \tag{4.12}$$

$$z_3 = (x_3 - m_3(x_1, x_2)) \cdot \exp(-s_3(x_1, x_2)) \tag{4.13}$$

$$\vdots \tag{4.14}$$

$$z_n = (x_n - m_n(x_1, x_2, \ldots, x_{n-1})) \cdot \exp(-s_n(x_1, x_2, \ldots, x_{n-1})) \tag{4.15}$$

2. **First Permutation Layer:**

   - Apply a permutation to the transformed variables to introduce dependencies across dimensions:

$$\mathbf{z}' = \mathbf{P}_1 \mathbf{z} \tag{4.16}$$

- For example, if $\mathbf{P}_1$ reverses the order, the result is:

$$\mathbf{z}' = [z_n, z_{n-1}, \ldots, z_1] \tag{4.17}$$

3. **Second MAF Layer:**

- For the first dimension of $\mathbf{z}'$:

$$y_1 = z'_1 \cdot \exp(-s_3) + m_3 \tag{4.18}$$

- For the other dimensions:

$$y_2 = (z'_2 - m_4(z'_1)) \cdot \exp(-s_4(z'_1)) \tag{4.19}$$

$$y_3 = (z'_3 - m_5(z'_1, z'_2)) \cdot \exp(-s_5(z'_1, z'_2)) \tag{4.20}$$

$$\vdots \tag{4.21}$$

$$y_n = (z'_n - m_{n+2}(z'_1, z'_2, \ldots, z'_{n-1})) \cdot \exp(-s_{n+2}(z'_1, z'_2, \ldots, z'_{n-1})) \tag{4.22}$$

4. **Second Permutation Layer:**

- Apply another permutation:

$$\mathbf{y}' = \mathbf{P}_2 \mathbf{y} \tag{4.23}$$

- For instance, if $\mathbf{P}_2$ reverses the order:

$$\mathbf{y}' = [y_n, y_{n-1}, \ldots, y_1] \tag{4.24}$$

5. **Final Output:**

- The final output in the latent space is:

$$\mathbf{z} = \mathbf{y}' = [y_n, y_{n-1}, \ldots, y_1] \tag{4.25}$$

Backward Pass

The backward pass reverses the transformations in the forward pass:

1. **Second Permutation Layer:**

$$\mathbf{y} = \mathbf{P}_2^{-1}\mathbf{y}' \tag{4.26}$$

If $\mathbf{P}_2$ is a reverse permutation:

$$\mathbf{y} = [y_1, y_2, \ldots, y_n] \tag{4.27}$$

2. **Second MAF Layer:**

- For the first dimension of $\mathbf{y}$:

$$z_1' = (y_1 - m_3) \cdot \exp(s_3) \tag{4.28}$$

- For the other dimensions:

$$z_2' = (y_2 - m_4(z_1')) \cdot \exp(s_4(z_1')) \tag{4.29}$$

$$z_3' = (y_3 - m_5(z_1', z_2')) \cdot \exp(s_5(z_1', z_2')) \tag{4.30}$$

$$\vdots \tag{4.31}$$

$$z_n' = (y_n - m_{n+2}(z_1', z_2', \ldots, z_{n-1}')) \cdot \exp(s_{n+2}(z_1', z_2', \ldots, z_{n-1}')) \tag{4.32}$$

3. **First Permutation Layer:**

$$\mathbf{z} = \mathbf{P}_1^{-1}\mathbf{z}' \tag{4.33}$$

If $\mathbf{P}_1$ is a reverse permutation:

$$\mathbf{z} = [z_1, z_2, \ldots, z_n] \tag{4.34}$$

4. **First MAF Layer:**

- For the first dimension:

$$x_1 = (z_1 - m_1) \cdot \exp(s_1) \tag{4.35}$$

- For the remaining dimensions:

$$x_2 = (z_2 - m_2(x_1)) \cdot \exp(s_2(x_1)) \tag{4.36}$$

$$x_3 = (z_3 - m_3(x_1, x_2)) \cdot \exp(s_3(x_1, x_2)) \tag{4.37}$$

$$\vdots \tag{4.38}$$

$$x_n = (z_n - m_n(x_1, x_2, \ldots, x_{n-1})) \cdot \exp(s_n(x_1, x_2, \ldots, x_{n-1})) \tag{4.39}$$

5. **Final Output:**

   - The recovered input vector is:

$$\mathbf{x} = [x_1, x_2, \ldots, x_n] \tag{4.40}$$

4.2.2.4 Training Process Overview

In this model, the input is the feature vector **x**, and the output is the log-probability derived from the transformed distribution. The model parameters are initialized using the Glorot Normal initializer [276], which ensures stability during training. Optimization is performed with the Adam algorithm, using an exponentially decreasing learning rate to achieve better convergence. The training process consists of the following steps:

1. **Initialization:**

   - Set up the initial parameters of the network.

2. **Forward Pass:**

   - For each input data point **x** in the dataset, pass it through the series of transformations to derive the corresponding latent representation **z**. - Calculate the log-likelihood of **x** using the base distribution and the Jacobian determinants

of the transformations.

3. **Loss Computation:**

   - The training objective is to minimize the negative log-likelihood of the dataset:

$$\mathscr{L} = -\frac{1}{N}\sum_{i=1}^{N} \log p_{\mathbf{x}}(\mathbf{x}^{(i)}) \tag{4.41}$$

4. **Backward Pass:**

   - Calculate the gradients of the loss function with respect to the parameters of the transformations using backpropagation. - Update the model parameters using a suitable optimization algorithm.

5. **Repeat:**

   - Continue iterating through the dataset, performing forward and backward passes until the model achieves satisfactory performance.

The proposed architecture balances simplicity and effectiveness, making it suitable for modeling intricate data distributions while remaining computationally efficient for training and inference. A comprehensive visualization of the architecture, which incorporates Masked Autoregressive Flows (MAF) and permutation layers, is provided in Figure 4.4.

4.2.3 Synthetic Dataset Generation

The goal of this process is to generate a synthetic dataset that mimics the characteristics of the original dataset using the normalizing flow model. This synthetic dataset is used to assign pseudolabels for outlier detection. The size of the generated dataset can be adjusted according to the needs of the application. For example, it can be smaller than or larger than the original dataset. In our experiments, we generated a synthetic dataset twice the size of the original dataset, since this approach provides a better representation of the minority class by increasing the number of available examples. The procedure for generating the synthetic dataset is described below:

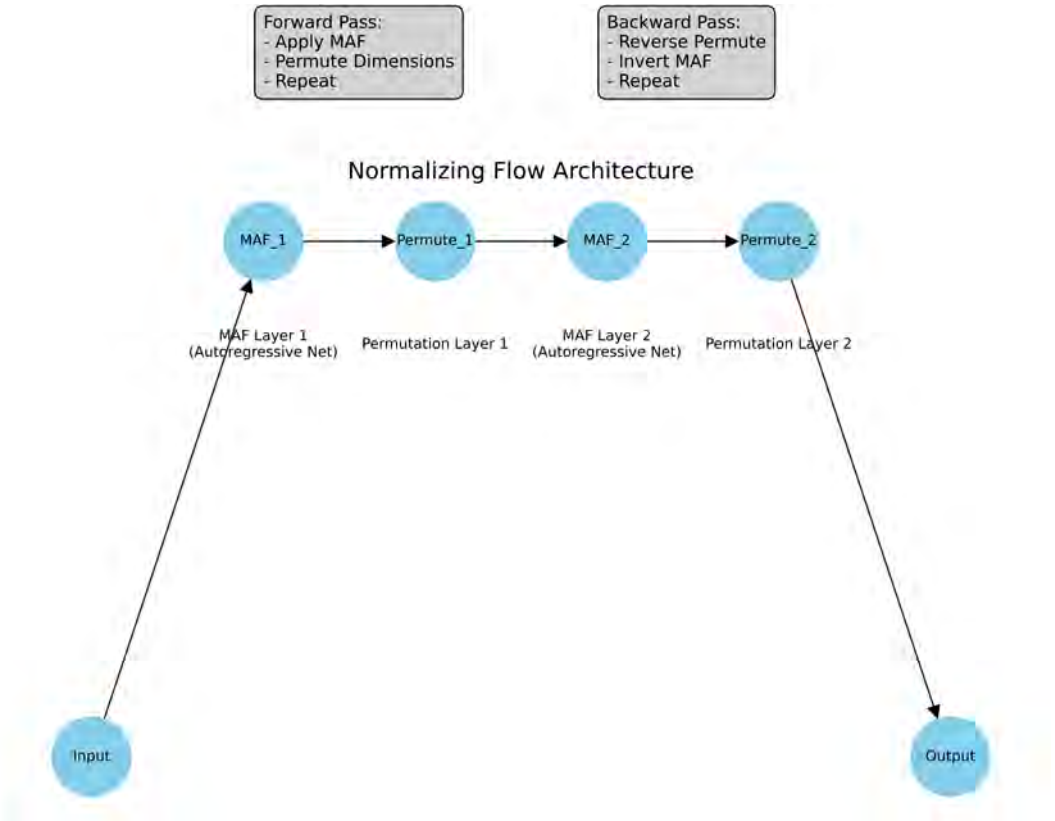

Figure 4.4: Overview of the Normalizing Flow architecture with Masked Autoregressive Flows (MAF) and Permutation Layers.

1. Sampling from the Base Distribution

Data points are sampled from the base distribution modeled by the normalizing flow:

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{4.42}$$

2. Applying the Inverse Transformations

The sampled points, represented as $\mathbf{z}$, are mapped back to the original data space by applying the inverse of the bijective transformations learned by the normalizing flow model.

$$\mathbf{x}_{\text{synthetic}} = f_{\text{chain}}^{-1}(\mathbf{z}) \tag{4.43}$$

where f_{chain}^{-1} represents the inverse of the sequence of transformations defined by the normalizing flow.

3. Generating the Synthetic Dataset

By iterating through the sampling and transformation steps, a synthetic dataset is constructed that reflects the probability distribution of the original dataset.

$$\mathbf{X}_{\text{synthetic}} = \{\mathbf{x}_{\text{synthetic},i}\}_{i=1}^{N} \tag{4.44}$$

4. Data Clipping

Finally, the values in the synthetic dataset are clipped to align with the range of the original dataset, ensuring that all values remain realistic while still capturing potential outliers.

4.2.4 Pseudo-label Generation

After the synthetic dataset is generated, pseudolabels are assigned using the probability densities computed by the normalizing flow model. These pseudolabels are critical for detecting outliers in an unsupervised manner and are subsequently used to train a decision tree for interpretability. The process involves the following steps:

1. Compute Probability Densities

For each data point $\mathbf{x}_{\text{synthetic},i}$ in the synthetic dataset, its probability density is calculated under the learned distribution. This is achieved by mapping $\mathbf{x}_{\text{synthetic},i}$ back through the sequence of invertible transformations in the normalizing flow to obtain $\mathbf{z}_i$, and then evaluating the density of $\mathbf{z}_i$ under the base distribution:

$$\log p_{\mathbf{x}}(\mathbf{x}_{\text{synthetic},i}) = \log p_z\big(f^{-1}(\mathbf{x}_{\text{synthetic},i})\big) + \log \left| \det \frac{\partial f^{-1}(\mathbf{x}_{\text{synthetic},i})}{\partial \mathbf{x}_{\text{synthetic},i}} \right| \tag{4.45}$$

Here:

- f^{-1} represents the inverse transformations applied to $\mathbf{x}_{\text{synthetic},i}$,
- $\log p_z$ is the log-probability under the base distribution,
- The Jacobian determinant $\det \frac{\partial f^{-1}(\mathbf{x}_{\text{synthetic},i})}{\partial \mathbf{x}_{\text{synthetic},i}}$ accounts for the change in volume during the transformations.

2. Thresholding for Outliers

To classify synthetic data points as inliers or outliers, a threshold is defined based

on their probability densities. Following the approach used in PyOD [79], a well-known library and benchmark for outlier detection, a specified percentage of data points (e.g. 2%) are identified as outliers. Data points are ranked by their probability densities, with those below the threshold labeled as outliers and those above it labeled as inliers. This method ensures consistency across models in benchmarks, including the proposed approach, enabling fair and unbiased comparisons.

- **Inliers:** Data points with high probability densities (above the threshold).
- **Outliers:** Data points with low probability densities (below the threshold).

The threshold can be expressed as:

$$\text{Threshold} = \text{percentile}\left(\{\log p_{\mathbf{x}}(\mathbf{x}_{\text{synthetic},i})\}_{i=1}^{N}, \alpha\right) \tag{4.46}$$

where α represents the percentage of points that should be classified as outliers.

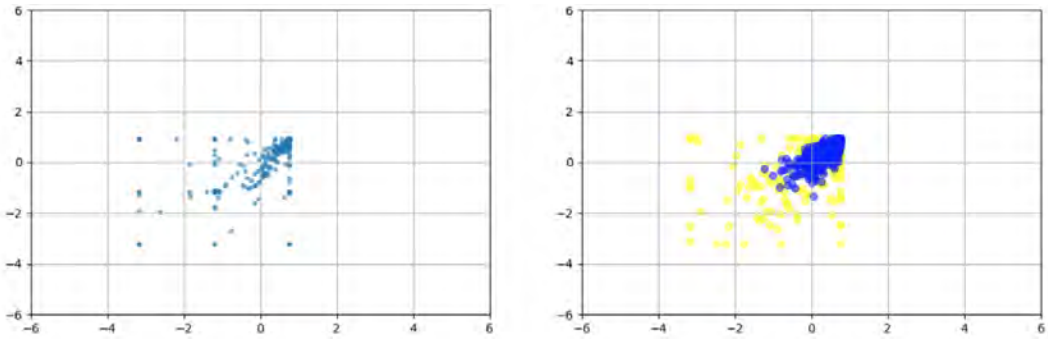

Figure 4.5: Two dimensions of the 33-dimensional Ionosphere dataset used to train a normalizing flow (left) and the synthetic dataset generated by the trained model (right). Applying an outlier percentage $\alpha = 0.36$ to the synthetic dataset, 64% of the points with the highest probability density are labeled as inliers (blue), and 36% with the lowest probability density are labeled as outliers (yellow).

By assigning pseudolabels, a labeled dataset is constructed, with outliers identified based on their likelihoods under the learned distribution. This labeled dataset can then be used to train a decision tree, allowing interpretability of the outlier detection process and identifying the features that contribute to each classification.

4.2.5 Decision Tree Training

The goal of this step is to use the pseudolabels generated by the normalizing flow model to train a decision tree. Decision trees are inherently interpretable, providing insight into the decision-making process by analyzing the structure of the tree and the importance of the features. Training a decision tree on the pseudo-labeled data creates a model capable of classifying the original dataset into inliers and outliers while offering clear explanations for its classifications.

1. Preparing the Training Data

To train the decision tree, the original dataset is paired with the pseudolabels from the previous step. The features of the dataset act as inputs to the decision tree, while the pseudolabels serve as the target output.

$$\text{Training Data} = (\mathbf{X}_{\text{synthetic}}, \mathbf{y}) \tag{4.47}$$

Here, $\mathbf{X}_{\text{synthetic}}$ represents the synthetic dataset, and $\mathbf{y}$ denotes the pseudolabels.

2. Training the Decision Tree

The decision tree classifier is trained on the prepared dataset, learning to classify data points based on the relationships between features and the pseudolabels. Standard training procedures for decision trees include:

- Identifying the best split points using the Gini impurity criterion.
- Pruning the tree to avoid overfitting and improve generalization.
- Evaluating the model performance using 5-fold cross-validation with the F1 Score as the evaluation metric.

4.2.6 Decision Tree Scoring

After training the decision tree, it is applied to the original dataset to classify each data point as an inlier or an outlier. The decision tree also provides interpretability by identifying the features that contribute the most significantly to the classifications.

$$\hat{y}_i = \text{DecisionTreeClassifier}(\mathbf{x}_i) \tag{4.48}$$

In the equation, $\hat{y}_i$ represents the predicted label (inlier or outlier) for the data point $\mathbf{x}_i$ in the original dataset.

4.2.7 Extracting Feature Contributions and Rules

During both the training and scoring phases, the feature importances can be extracted to determine which features have the greatest influence on the classification of inliers and outliers. The feature importance is calculated on the basis of the reduction in impurity achieved by a feature at each split point in the decision tree.

$$\text{Feature Importance} = \sum_{t \in \text{Splits}} \Delta\text{Impurity}(t, \text{feature})$$

Here, $\Delta\text{Impurity}(t, \text{feature})$ represents the reduction in impurity attributed to the feature at split t.

Due to the transparent nature of decision trees, feature contributions can be analyzed both globally (across the entire training or scoring dataset) and locally (for individual data points). In addition, decision rules are derived to further enhance the interpretability of the proposed method.

4.3 Data and Experiments

This section provides an overview of the datasets and experiments conducted to validate the proposed methodology. It describes the datasets, including their sources, characteristics, and preprocessing steps, followed by a description of the experimental setup, training process, evaluation metrics, and parameters used.

4.3.1 Datasets

The proposed approach was evaluated using 17 benchmark datasets from the standardized collection provided by the PyOD library (Python Outlier Detection) [79], a well-established resource in outlier detection research. These datasets provide a consistent and unbiased benchmark, enabling fair model comparisons. All datasets are publicly accessible through the PyOD GitHub repository.

To ensure fairness, identical datasets and evaluation metrics were used across all models, ensuring that the results accurately reflect each model's inherent capa-

bilities without external biases. Table 4.2 summarizes the datasets used, including the number of samples, dimensions, and the percentage of outliers.

Table 4.2: Summary of datasets used in the experiments, detailing sample size, dimensionality, outlier percentage, application domain, and data type.

Dataset	#Samples	#Dimensions	%Outliers	Domain	Data Type
arrhythmia	452	274	14.60	Healthcare	Tabular
cardio	1831	21	9.61	Healthcare	Tabular
glass	214	9	4.21	Forensics	Tabular
ionosphere	351	33	35.90	Space Physics	Tabular
letter	1600	32	6.25	Computer Vision	Tabular
lympho	148	18	4.05	Healthcare	Tabular
mnist	7603	100	9.21	Computer Vision	Image
musk	3062	166	3.17	Chemistry	Tabular
optdigits	5216	64	2.88	Computer Vision	Tabular
pendigits	6870	16	2.27	Computer Vision	Tabular
pima	768	8	34.90	Healthcare	Tabular
satellite	6435	36	31.64	Remote Sensing	Tabular
satimage-2	5803	36	1.22	Remote Sensing	Tabular
shuttle	49097	9	7.15	Aerospace	Tabular
vertebral	240	6	12.50	Healthcare	Tabular
vowels	1456	12	3.43	Linguistics	Tabular
wbc	378	30	5.56	Healthcare	Tabular

4.3.1.1 Preprocessing Steps

Before training and evaluation, all datasets were normalized, with features standardized to have zero mean and unit variance. This step ensured the normalizing flow model learned a well-behaved probability distribution. The dataset was then divided into 70% for training and 30% for evaluation.

4.3.2 Experimental Setup

This subsection details the experimental setup, including the training process, evaluation metrics, and parameters used.

4.3.2.1 Training Process

The training process of the proposed methodology comprises the following steps:

1. **Training the Normalizing Flow Model**: The normalizing flow model was trained on the training data to learn the underlying data distribution.

2. **Generating Synthetic Data**: Synthetic data were generated by sampling the base distribution and applying the learned transformations.

3. **Pseudo-label Generation**: The probability densities for the synthetic data points were calculated and pseudolabels were assigned based on a predefined threshold (for example, the bottom 2% labeled as outliers).

4. **Training the Decision Tree**: A decision tree classifier was trained on the pseudolabeled synthetic dataset to distinguish between inliers and outliers.

5. **Application to the Original Dataset**: The trained decision tree was applied to the original dataset to classify data points and extract feature contributions and decision rules for interpretability.

4.3.2.2 Parameters Used

The key hyperparameter choices are summarized below. These were selected to balance simplicity and performance, ensuring effective results without unnecessary complexity. Details on additional parameters are available in the accompanying code repository.

- **Normalizing Flow Model Architecture**:
  - Number of layers: 2
  - Hidden units: [128, 128]
  - Activation function: tanh

  This configuration reduces overfitting while capturing key data distribution patterns.
- **Normalizing Flow Model Training**:
  - Optimizer: Adam

  - Learning rate: 0.001 with exponential decay
  - Parameter initializer: Glorot Normal
  - Number of epochs: 300

  These values were chosen on the basis of initial experiments and were found sufficient for convergence while maintaining computational efficiency.

- **Decision Tree Tuning**:
  - 5-fold cross-validation with the F1 score was used as the evaluation metric to select the best hyperparameters.

4.3.2.3 Evaluation Metrics

To comprehensively evaluate the proposed approach, we used multiple metrics that capture different aspects of performance, especially important in the detection of outliers due to class imbalance.

Precision

Precision measures the proportion of correctly identified outliers (true positives) among all predicted outliers:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4.49}$$

This metric is critical in scenarios where false positives are costly, such as fraud detection or medical diagnosis, as it reduces unnecessary actions.

Recall

Recall quantifies the proportion of true positives among all actual outliers:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4.50}$$

High recall ensures that most outliers are detected, which is essential in high-stakes contexts like identifying fraud or diseases.

F1 Score

The F1 score, the harmonic mean of Precision and Recall, balances the trade-off between false positives and false negatives.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.51}$$

F1 is particularly useful for imbalanced datasets but does not account for true negatives, making it less suitable for applications requiring inlier classification accuracy.

Matthews Correlation Coefficient (MCC)

MCC evaluates overall performance by incorporating all elements of the confusion matrix.

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{4.52}$$

MCC is well-suited for imbalanced datasets, providing a holistic view of the model's ability to handle both outliers and inliers, making it more comprehensive than the F1 score.

Combining the F1 score and MCC provides a thorough evaluation: The F1 score ensures effective outlier detection by balancing Precision and Recall, while MCC assesses overall accuracy, including inlier classification. Together, these metrics enable a nuanced understanding of the model's strengths and weaknesses, tailored to application-specific needs.

4.3.2.4 Model Comparison

The proposed method was evaluated against 23 established models selected from the PyOD library (Python Outlier Detection) [79], a widely recognized benchmark for outlier detection research. These models encompass various approaches, including statistical, proximity-based, probabilistic, and machine learning techniques, providing a comprehensive comparison. Ensemble methods, such as Feature Bagging, were excluded to ensure a fair comparison since the proposed method is not ensemble-based. The models also include recent advances, such as the Deep Isolation Forest based on deep learning [277].

Below is an overview of the competing algorithms.

1. Angle-Based Outlier Detection (ABOD) [27] detects outliers by assessing the variance in weighted cosine scores between an observation and its neighbors. Higher variance indicates a higher likelihood of being an outlier.

2. Cluster-Based Local Outlier Factor (CBLOF) [65] assigns outlier scores based on the size of the cluster to which a point belongs and its distance from larger clusters.

3. Histogram-Based Outlier Score (HBOS) [23] estimates data distributions using feature histograms, assuming feature independence.

4. Isolation Forest (IForest) [72] isolates outliers by recursively partitioning data using randomly selected features and values.

5. Deep Isolation Forest (Deep IForest) [277] extends IForest with deep learning-based representations for non-linear isolation.

6. K-Nearest Neighbors (KNN) [26] detects outliers by measuring the distance to the k-th nearest neighbor.

7. Average K-Nearest Neighbors (Average KNN) calculates the average distance to the k-nearest neighbors.

8. Median K-Nearest Neighbors (Median KNN) uses the median distance to the k-nearest neighbors, making it less sensitive to outliers.

9. Local Outlier Factor (LOF) [22] measures the density deviation of a point compared to its neighbors, identifying points with lower density as outliers.

10. Minimum Covariance Determinant (MCD) [69] identifies outliers using the Mahalanobis distance, assuming Gaussian data distributions.

11. One-Class SVM (OCSVM) [68] maps data into a higher-dimensional space and separates normal and anomalous points using a hyperplane.

12. Principal Component Analysis (PCA) [70] projects data on principal components to detect deviations as outliers.

13. Kernel PCA (KPCA) [278] applies kernel functions to perform non-linear dimensionality reduction for anomaly detection.

14. Empirical Cumulative Distribution (ECOD) [58] uses empirical cumulative distribution functions to estimate tail probabilities and detect outliers.

15. Gaussian Mixture Model (GMM) [279] fits data using a mixture of Gaussian distributions and identifies outliers based on low-probability points.

16. Isolation Using Nearest Neighbors Ensemble (INNE) [280] employs ensembles of nearest-neighbor distances to isolate outliers.

17. Copula-Based Outlier Detection (COPOD) [281] uses copulas to model dependencies between variables and calculates tail probabilities to detect anomalies.

18. One-Time Sampling [282] randomly samples a subset of data and measures the minimum distance from each point to the sampled subset.

19. Local Unsupervised Outlier Detection Using Robust Statistics (LUNAR) [283] leverages graph neural networks to model nearest-neighbor relationships for outlier detection.

20. Kernel Density Estimation (KDE) [284] estimates local densities using variable kernels and flags points with low densities as outliers.

21. Locally Mahalanobis Distance Detection (LMDD) [285] detects anomalies based on the dissimilarities measured by the Mahalanobis distance.

22. Autoencoder [286] identifies outliers based on a high reconstruction error from a compressed representation.

23. Deep Support Vector Data Description (DeepSVDD) [287] minimizes the volume of a hypersphere enclosing normal data in a latent space, classifying distant points as outliers.

This comprehensive comparison ensures a thorough evaluation of the proposed method against a wide range of state-of-the-art techniques, highlighting its effectiveness in diverse scenarios.

4.3.2.5 Statistical Analysis: Friedman Test and Post-Hoc Holm Procedure

To evaluate the statistical significance of performance differences among outlier detection models, we employed the Friedman test, followed by the post-hoc Holm

procedure. This robust framework ensures that observed differences are meaningful and not due to random variation or multiple comparisons. Widely used in machine learning, the Friedman test identifies overall performance differences across algorithms, while the Holm procedure pinpoints specific pairwise differences, controlling for Type I errors. Together, they provide a rigorous and interpretable evaluation of model rankings across datasets, critical for confirming algorithm superiority in diverse real-world applications.

Friedman Test: Detecting Overall Differences

The Friedman test ranks algorithms for each dataset and evaluates whether their rank differences are statistically significant. For k algorithms evaluated on N datasets, the average rank R_i for algorithm i is:

$$R_i = \frac{1}{N} \sum_{j=1}^{N} r_{ij}$$

where r_{ij} is the rank of algorithm i on dataset j. The test statistic χ_F^2 is:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left(\sum_{i=1}^{k} R_i^2 - \frac{k(k+1)^2}{4} \right)$$

with χ_F^2 following a chi-squared distribution with $k-1$ degrees of freedom.

Post-Hoc Holm Procedure: Pinpointing Significant Comparisons

If the Friedman test reveals significant differences, the Holm procedure determines which algorithm pairs differ meaningfully. The p-values for pairwise comparisons are ordered as $p_1, p_2, \ldots, p_m$, and adjusted significance levels α_i are calculated as:

$$\alpha_i = \frac{\alpha}{m - i + 1}$$

where α is the overall significance level (e.g., 0.05). Null hypotheses for comparisons are rejected iteratively if $p_i < \alpha_i$.

This combined approach is invaluable for benchmarking machine learning models, ensuring robust conclusions about algorithm performance while avoiding

misleading results due to chance. It is particularly critical in fields like outlier detection, where class imbalance and dataset diversity can obscure true performance differences.

4.3.2.6 Further Implementation Details

Comprehensive implementation details are available in a publicly accessible repository, ensuring reproducibility and facilitating further research. The experiments extensively used the PyOD library [79] for the detection of outliers.

4.4 Results and Discussion

This section presents the experimental results, including comparisons with baseline models, and discusses the advantages and limitations of the proposed approach. Insights derived from the feature importance analysis are also included. The proposed method was evaluated on 17 datasets and compared with 23 baseline models using metrics such as the F1 score, the Matthews correlation coefficient (MCC), precision, and recall. Statistical analysis using the Friedman test and post hoc analysis showed no significant performance differences between our method and the baseline models. However, the proposed method demonstrated a significantly higher level of interpretability compared to competing models.

4.4.1 Training Losses

The training loss curves (Figure 4.6) in the 17 datasets demonstrate the ability of the normalizing flow model to learn data distributions effectively. For datasets such as arrhythmia.mat, cardio.mat, and satellite.mat, the loss smoothly decreased, indicating efficient learning of the properties of the data.

Some datasets, including mnist.mat and shuttle.mat, exhibited initial fluctuations in loss values, which stabilized as training progressed, reflecting the model's eventual success in learning the distributions. However, datasets like lympho.mat and vertebral.mat showed persistent fluctuations throughout training, suggesting challenges in consistently capturing their underlying distributions.

Overall, while the normalizing flow model performed well for most datasets, its stability varied, particularly for more complex or irregular distributions.

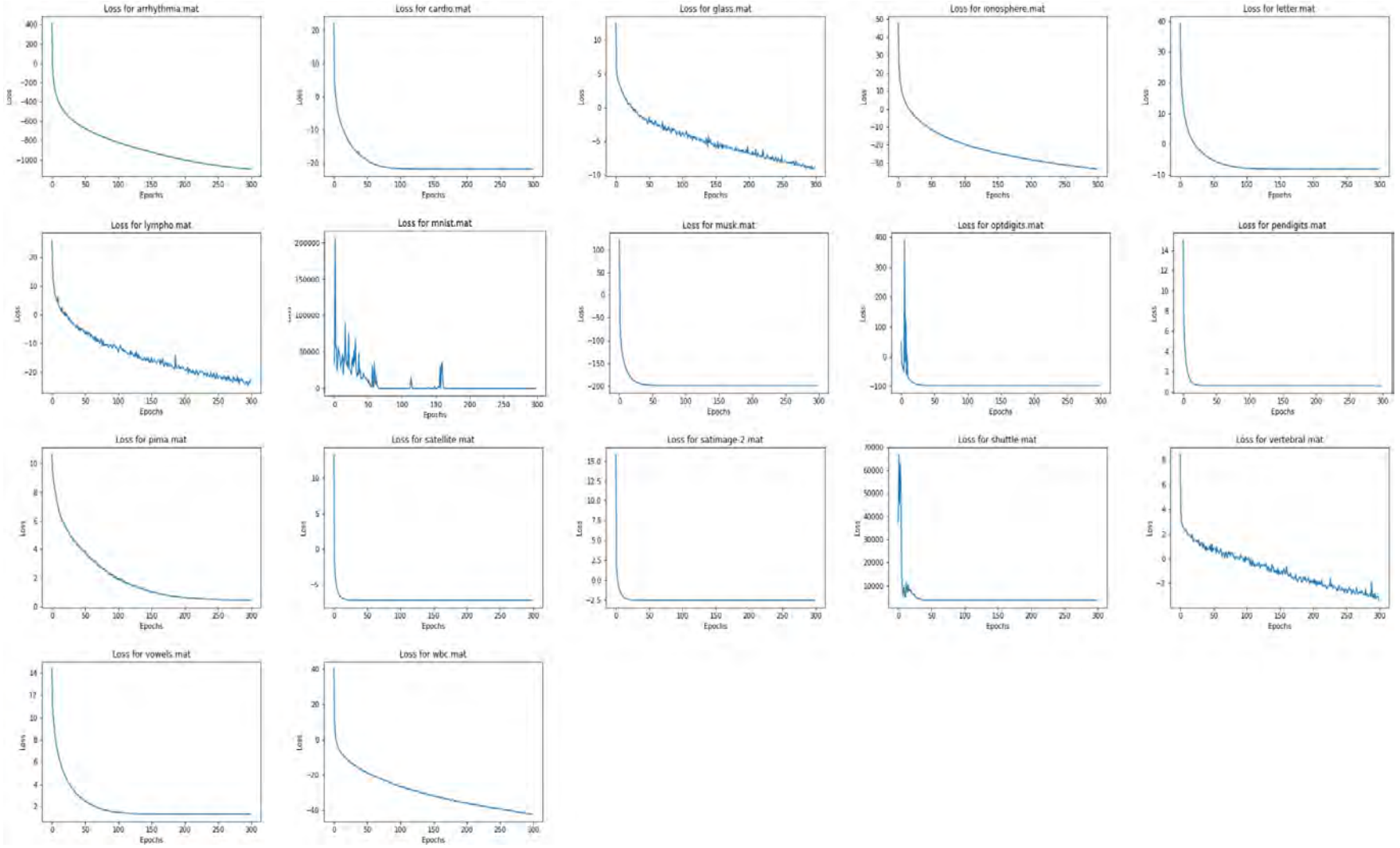

Figure 4.6: Training losses (negative log-likelihood) for all 17 datasets using the normalizing flow model over 300 epochs. The negative log-likelihood measures how well the model fits the data distribution during training.

4.4.2 Execution Times

When assessing the computational efficiency of outlier detection models, it is important to distinguish between training time and scoring time. Training is typically performed offline and can afford to be more time-intensive unless frequent retraining is required. In contrast, test time directly impacts real-time performance, making it a critical metric for practical applications. The execution times for the proposed method and 23 baseline models were measured in 17 datasets, as summarized in Tables 4.3 and 4.4.

4.4.2.1 Training Time

The training time results, detailed in Table 4.3, reveal that the proposed method has significantly longer training times compared to most baseline models. For example, on the mnist dataset, the proposed method required 312.61 seconds, while simpler models such as HBOS and ECOD completed the training in just 0.04 and 0.09 seconds, respectively. Similarly, on the shuttle dataset, the training time for the proposed method was 1839.48 seconds, compared to only 0.03 seconds for HBOS and 0.08 seconds for COPOD.

The longer training times are attributed to the computational demands of normalizing flows, which involve multiple transformation stages to accurately model the data distribution. Despite these longer training times, the impact on real-time performance is minimal in most applications, as training typically occurs offline.

However, in scenarios that require frequent model retraining, the extended training time could pose challenges. In such cases, strategies such as model optimization or the adoption of faster alternative methods may need to be explored to mitigate this limitation.

4.4.2.2 Scoring Time

As shown in Table 4.4, the proposed method demonstrated consistently fast scoring times in all datasets. For example, on the cardio dataset, the scoring time for the proposed method was only 0.0002 seconds, significantly outperforming models such as IForest and DeepSVDD, which required 0.0519 and 0.0724 seconds, re-

Table 4.3: Training time (in seconds) for each dataset and model

Model / Dataset	arrhythmia	cardio	glass	ionosphere	letter	lympho	mnist	musk	optdigits	pendigits	pima	satellite	satimage-2	shuttle	vertebral	vowels	wbc
ABOD	1.3870	0.2860	0.0250	0.0709	0.2613	0.0276	1.0323	0.4549	0.1646	0.3213	0.0901	0.8312	0.7587	8.4604	0.0276	0.1745	0.0768
AutoEncoder	7.3929	10.0020	4.4648	5.2891	9.6240	4.4244	35.4750	34.8139	5.0485	14.9577	6.4066	27.8738	24.6798	174.6256	4.4241	8.8937	5.0592
Average KNN	0.0067	0.0078	0.0016	0.0029	0.0076	0.0025	0.0971	0.0364	0.0083	0.0123	0.0047	0.0485	0.0336	3.1134	0.0015	0.0147	0.0030
CBLOF	1.1793	0.1023	0.0360	0.0426	0.0911	0.0398	0.3140	0.1373	0.0909	0.0759	0.0608	0.1776	0.1886	0.2029	0.0388	0.0667	0.0559
COPOD	0.1294	0.0065	0.0017	0.0027	0.0069	0.0017	0.0976	0.0982	0.0123	0.0124	0.0023	0.0302	0.0272	0.0784	0.0016	0.0044	0.0026
Deep IForest	1.6592	3.1760	1.0359	1.2609	2.8553	0.9554	13.6051	7.3897	5.6608	6.8260	1.6762	10.7388	9.1211	84.2851	1.0921	2.5427	1.2359
DeepSVDD	4.0572	7.0480	3.5825	3.5276	6.5088	3.0132	21.4746	18.6316	5.1124	5.5522	4.6178	18.2599	16.7890	120.2405	3.2616	6.2256	5.4860
ECOD	0.0115	0.0058	0.0016	0.0025	0.0068	0.0016	0.0864	0.0787	0.0103	0.0107	0.0022	0.0277	0.0250	0.0612	0.0017	0.0046	0.0026
GMM	0.0234	0.0069	0.0046	0.0051	0.0071	0.0051	0.0391	0.0326	0.0086	0.0459	0.0048	0.0149	0.0141	0.1987	0.0058	0.0056	0.0049
HBOS	2.6218	0.0060	0.0024	0.0101	0.0089	0.0043	0.0449	0.1485	0.0135	0.0459	0.0024	0.0160	0.0148	0.0253	0.0018	0.0043	0.0070
IForest	0.3243	0.3431	0.2027	0.3093	0.3384	0.2885	0.6086	1.2616	0.2769	0.3297	0.2195	0.4777	0.4556	1.3143	0.2086	0.2473	0.2973
INNE	0.1930	0.2029	0.1429	0.1526	0.2007	0.1425	0.5268	0.9040	0.2826	0.3161	0.1758	0.3964	0.3497	1.4753	0.1444	0.2009	0.1490
KDE	0.0623	0.1753	0.0025	0.0101	0.1886	0.0017	7.3730	3.6825	0.0135	0.0506	0.0239	3.3609	2.9227	112.0954	0.0028	0.1088	0.0110
KNN	0.0089	0.0070	0.0016	0.0032	0.0125	0.0028	0.0912	0.0454	0.0066	0.0108	0.0048	0.0572	0.0445	3.6489	0.0017	0.0141	0.0058
KPCA	0.1165	0.6233	0.0435	0.0758	0.4992	0.0301	21.5747	0.7303	0.0854	0.2144	0.1609	12.9962	9.8443	6178.2920	0.0461	0.4060	0.0808
LMDD	3.8029	5.2418	0.2272	0.4900	5.2984	0.1868	442.5049	52.1393	3.0060	6.3325	0.7654	95.7511	75.5936	1573.3373	0.2361	2.2637	0.5350
LOF	0.0074	0.0192	0.0025	0.0045	0.0130	0.0030	0.1038	0.0532	0.0071	0.0126	0.0075	0.0608	0.0451	5.6946	0.0025	0.0245	0.0085
LUNAR	2.9595	3.8569	1.5468	2.1222	3.5872	1.4558	19.4320	8.9196	4.1677	3.8432	2.1818	15.7355	13.7962	154.0852	1.5121	3.2885	1.8835
MCD	0.7244	0.5182	0.0346	0.0789	1.2788	0.0392	2.7357	0.2971	0.6234	0.4355	0.5134	2.9004	2.4955	15.5612	0.0679	0.7790	0.0628
Median KNN	0.0071	0.0081	0.0017	0.0032	0.0083	0.0023	0.0950	0.0449	0.0082	0.0117	0.0065	0.0490	0.0375	3.1561	0.0017	0.0145	0.0034
OCSVM	0.0210	0.2362	0.0052	0.0156	0.2026	0.0046	4.6631	1.3762	0.0370	0.0445	0.0472	3.1077	2.1939	169.0055	0.0066	0.1666	0.0142
PCA	0.0445	0.0041	0.0016	0.0032	0.0052	0.0015	0.0756	0.0659	0.0123	0.0145	0.0016	0.0171	0.0134	0.0247	0.0027	0.0021	0.0022
Proposed	45.7236	59.5267	15.7126	22.3013	55.4959	15.1045	312.6061	580.3967	43.9125	60.8559	27.3973	288.5200	243.6259	1839.4829	16.5825	45.6393	21.1687
Sampling	0.0039	0.0011	0.0006	0.0008	0.0013	0.0007	0.0098	0.0083	0.0017	0.0020	0.0010	0.0033	0.0029	0.0200	0.0007	0.0011	0.0009

spectively. Across all datasets, the proposed method consistently achieved one of the lowest scoring times, frequently below 0.0002 seconds.

This exceptional efficiency during testing makes the proposed approach well suited for real-time applications requiring low-latency anomaly detection. Scenarios such as fraud detection and network security, where immediate responses to anomalies are critical, particularly benefit from the rapid testing capabilities of our method.

4.4.3 F1 Results

This section evaluates the performance of the proposed method using the F1 score in various datasets. Key analyses include the correlation of the characteristics of the dataset with the performance of F1, the raw F1 scores for each dataset and model (Table 4.5), the Friedman ranking (Table 4.6) and post hoc comparisons with the Holm adjustment (Table 4.6).

Table 4.4: Scoring time (in seconds) for each dataset and model

Model / Dataset	arrhythmia	cardio	glass	ionosphere	letter	lympho	mnist	musk	optdigits	pendigits	pima	satellite	satimage-2	shuttle	vertebral	vowels	wbc
ABOD	0.0344	0.1068	0.0106	0.0294	0.0951	0.0114	1.6072	0.4795	0.1636	0.1979	0.0374	0.4731	0.4349	4.0354	0.0115	0.0725	0.0316
AutoEncoder	0.0506	0.0661	0.0454	0.0497	0.0547	0.0445	0.1289	0.1306	0.1296	0.1330	0.0520	0.1151	0.1060	0.5683	0.0440	0.0598	0.0481
Average KNN	0.0356	0.1077	0.0073	0.0194	0.0967	0.0078	1.3054	0.4183	0.1414	0.1754	0.0242	0.4190	0.3887	2.9981	0.0085	0.0502	0.0197
CBLOF	0.0031	0.0022	0.0014	0.0014	0.0021	0.0015	0.0075	0.0049	0.0020	0.0018	0.0015	0.0035	0.0033	0.0065	0.0016	0.0015	0.0018
COPOD	0.0178	0.0102	0.0014	0.0030	0.0121	0.0013	0.1414	0.1279	0.0120	0.0124	0.0028	0.0443	0.0408	0.0937	0.0013	0.0063	0.0031
Deep IForest	0.4119	1.0706	0.2807	0.3419	0.9631	0.2511	4.1522	4.0614	1.6530	2.2549	0.5189	3.1614	2.8998	28.3132	0.2841	0.8474	0.3816
DeepSVDD	0.0517	0.0724	0.0473	0.0500	0.0573	0.0458	0.1335	0.1310	0.1354	0.1386	0.0523	0.2288	0.1113	0.5862	0.0495	0.0644	0.0502
ECOD	0.0156	0.0072	0.0013	0.0027	0.0080	0.0013	0.1286	0.1221	0.0125	0.0125	0.0021	0.0403	0.0352	0.0908	0.0013	0.0043	0.0027
GMM	0.0012	0.0008	0.0004	0.0004	0.0009	0.0004	0.0035	0.0038	0.0008	0.0039	0.0004	0.0013	0.0010	0.0030	0.0004	0.0008	0.0004
HBOS	0.0010	0.0004	0.0002	0.0042	0.0006	0.0002	0.0049	0.0049	0.0044	0.0039	0.0002	0.0022	0.0020	0.0034	0.0002	0.0006	0.0002
IForest	0.0493	0.0519	0.0395	0.0412	0.0517	0.0413	0.1253	0.1703	0.0442	0.0518	0.0435	0.0850	0.0808	0.2767	0.0397	0.0490	0.0406
INNE	0.0645	0.0695	0.0444	0.0497	0.0729	0.0454	0.1988	0.3342	0.1159	0.1236	0.0520	0.6558	0.1536	0.6086	0.0515	0.0694	0.0477
KDE	0.0250	0.0743	0.0010	0.0042	0.0783	0.0006	3.2238	1.8549	0.0044	0.0236	0.0103	1.4418	1.2522	47.7514	0.0010	0.0429	0.0050
KNN	0.0253	0.0789	0.0049	0.0130	0.0754	0.0048	1.1985	0.4637	0.0901	0.1375	0.0172	0.3560	0.3300	2.5416	0.0054	0.0358	0.0142
KPCA	0.0314	0.1470	0.0201	0.0324	0.1309	0.0142	2.4104	0.1240	0.0291	0.0461	0.0470	1.6414	1.1387	52.1661	0.0144	0.1037	0.0286
LMDD	0.8588	1.1969	0.0523	0.1283	1.1771	0.0435	76.6542	9.7246	0.6045	1.2623	0.2222	17.2383	14.1535	249.4734	0.0589	0.6146	0.1365
LOF	0.0030	0.0055	0.0011	0.0024	0.0052	0.0014	0.0458	0.0224	0.0030	0.0049	0.0092	0.0281	0.0188	2.4441	0.0009	0.0088	0.0032
LUNAR	0.0110	0.0108	0.0016	0.0045	0.0096	0.0034	0.0832	0.0489	0.0064	0.0088	0.0042	0.0434	0.0392	0.4592	0.0014	0.0107	0.0041
MCD	0.0107	0.0007	0.0004	0.0005	0.0009	0.0004	0.0185	0.0038	0.0010	0.0010	0.0003	0.0022	0.0025	0.0013	0.0007	0.0004	0.0004
Median KNN	0.0461	0.1439	0.0095	0.0257	0.1254	0.0102	1.3722	0.5684	0.1547	0.2027	0.0334	0.4797	0.4450	3.5556	0.0098	0.0655	0.0254
OCSVM	0.0047	0.0406	0.0015	0.0048	0.0328	0.0010	1.3763	0.4682	0.0111	0.0135	0.0140	0.8773	0.7336	24.0047	0.0018	0.0339	0.0040
PCA	0.0124	0.0008	0.0002	0.0004	0.0012	0.0002	0.0142	0.0129	0.0027	0.0025	0.0002	0.0038	0.0036	0.0049	0.0002	0.0003	0.0003
Proposed	0.0002	0.0002	0.0001	0.0001	0.0002	0.0001	0.0023	0.0019	0.0002	0.0002	0.0001	0.0006	0.0005	0.0019	0.0001	0.0002	0.0001
Sampling	0.0009	0.0004	0.0002	0.0003	0.0005	0.0002	0.0039	0.0035	0.0007	0.0008	0.0003	0.0015	0.0014	0.0037	0.0002	0.0003	0.0003

4.4.3.1 Correlation of Dataset Characteristics with F1 Performance

The correlation analysis (Figure 4.7) highlights how the characteristics of the specific dataset affect the F1 performance of the proposed method. The most influential factors include the number of outliers (correlation: 0.36) and the number of samples (correlation: 0.34), indicating improved performance in datasets with more outliers and larger sample sizes. The percentage of outliers shows a moderate correlation (0.26), while the dimensionality has a weaker impact (0.13).

These findings suggest that the proposed method is particularly effective for larger datasets with a higher prevalence of outliers, as demonstrated by its strong performance on datasets like musk, shuttle, and satimage-2 (Table 4.5).

4.4.3.2 Raw F1 Results and Dataset-Specific Insights

A detailed examination of the raw F1 scores (Table 4.5) reveals the following patterns:

- **High Performers**: The method achieves its best F1 scores in datasets such

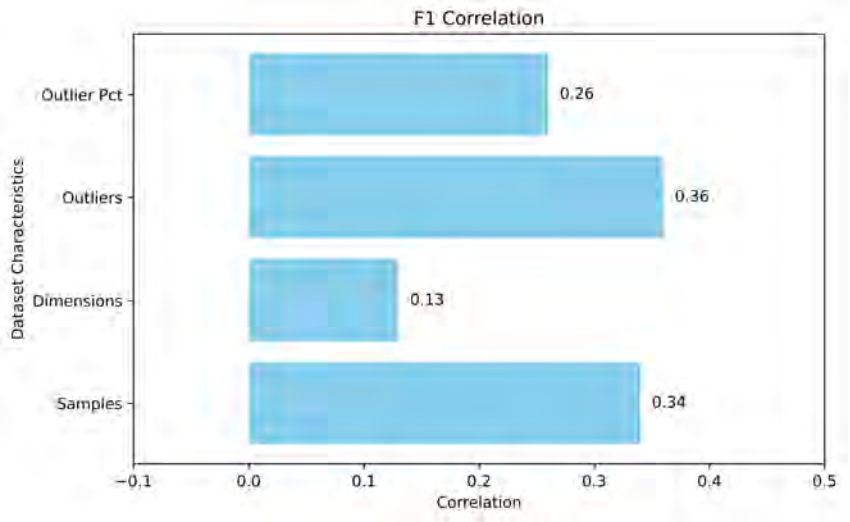

Figure 4.7: Correlations between the F1 score and different dataset characteristics.

as musk (F1: 0.8333), shuttle (F1: 0.7811), and satimage-2 (F1: 0.7119). These datasets are characterized by large sample sizes and a substantial number of outliers. For example, the shuttle dataset contains 49,097 samples with 3,510 outliers, providing ample data for the model to learn outlier patterns effectively.

- **Low Performers**: Performance is lower on datasets such as mnist (F1: 0.0103), and particularly poor on lympho (F1: 0.0) and vertebral (F1: 0.0). The mnist dataset, containing image data, presents challenges due to the instability of the model during training, as evidenced by spiky loss curves (Figure 4.6). Meanwhile, the small sample sizes of lympho and vertebral limit the model's ability to generalize effectively.

4.4.3.3 Friedman Ranking and Holm's post hoc analysis

The Friedman ranking (Table 4.6) places the proposed method 4th among 24 models, demonstrating its consistent performance across diverse datasets. In particular, the method outperforms well-established models such as PCA, KNN, and HBOS, highlighting its robustness and reliability. However, IForest and OCSVM achieve higher rankings, indicating that while the proposed method is among the best performers, there is potential for further optimization.

The results also emphasize the versatility of the proposed approach, which ef-

Table 4.5: F1 Score: Raw results per dataset and model

Model / Dataset	arrhythmia	cardio	glass	ionosphere	letter	lympho	mnist	musk	optdigits	pendigits	pima	satellite	satimage-2	shuttle	vertebral	vowels	wbc
ABOD	0.2941	0.1923	0.1667	0.2105	0.4416	0.3333	0.3485	0.0189	0.0459	0.0467	0.2991	0.2174	0.1798	0.1904	0.0000	0.3509	0.5000
AutoEncoder	0.3478	0.3885	0.0000	0.7143	0.3019	0.2857	0.3736	0.0000	0.5152	0.2000	0.4691	0.4564	0.4286	0.9034	0.1333	0.4800	0.3529
Average KNN	0.3000	0.2927	0.1250	0.3846	0.4471	0.6667	0.4009	0.1986	0.0087	0.0813	0.2752	0.3363	0.1471	0.2214	0.1053	0.4359	0.4444
CBLOF	0.3636	0.4545	0.1667	0.2105	0.2472	0.3077	0.4193	0.5124	0.0263	0.0892	0.3158	0.3503	0.2341	0.2917	0.0000	0.2909	0.5217
COPOD	0.4103	0.4828	0.1667	0.3396	0.0732	0.7273	0.2207	0.4561	0.0211	0.1781	0.3571	0.3914	0.1921	0.8378	0.0000	0.0333	0.4706
Deep IForest	0.2941	0.3636	0.0000	0.2128	0.1250	0.6000	0.4273	0.5000	0.0480	0.2687	0.0426	0.0066	0.3619	0.0678	0.0000	0.0000	0.0000
DeepSVDD	0.3636	0.4786	0.0000	0.5833	0.1034	0.5714	0.2654	0.0562	0.5538	0.2576	0.5542	0.5679	0.8108	0.5817	0.0000	0.0690	0.1538
ECOD	0.4390	0.4576	0.1429	0.3774	0.0988	0.7273	0.1822	0.4865	0.0296	0.2267	0.2500	0.3744	0.1702	0.8306	0.2727	0.1333	0.4706
GMM	0.2621	0.4228	0.1333	0.6000	0.3038	0.4000	0.3919	0.3367	0.0090	0.1039	0.3509	0.3206	0.1919	0.7981	0.1000	0.3659	0.5000
HBOS	0.5455	0.4318	0.1333	0.0370	0.1750	0.3077	0.0786	0.4247	0.2749	0.2462	0.3966	0.4520	0.2892	0.8683	0.0000	0.1250	0.4800
IForest	0.5455	0.4842	0.1667	0.2000	0.1149	0.4000	0.2648	0.5299	0.0263	0.2619	0.3793	0.4567	0.2667	0.8315	0.0000	0.1852	0.5455
INNE	0.2449	0.4000	0.1176	0.4444	0.2785	0.6667	0.3581	0.4583	0.0088	0.2101	0.2909	0.3282	0.1929	0.8345	0.0952	0.3429	0.4000
Median KNN	0.3000	0.2787	0.1333	0.3529	0.4186	0.6154	0.4194	0.2628	0.0263	0.0741	0.2617	0.3295	0.1493	0.2144	0.1053	0.4250	0.4211
KDE	0.2564	0.2534	0.2609	0.8172	0.1283	0.1633	0.1576	0.0571	0.0577	0.1511	0.3667	0.3647	0.1514	0.8376	0.0909	0.3400	0.1277
KNN	0.3636	0.2626	0.1818	0.1622	0.3562	0.4000	0.4051	0.2435	0.0274	0.1118	0.3761	0.3657	0.2162	0.2130	0.0000	0.3830	0.5217
KPCA	0.2564	0.1803	0.1429	0.5676	0.1069	0.1633	0.1575	0.0693	0.0547	0.0437	0.5426	0.4747	0.0216	0.4210	0.1772	0.0706	0.1000
LMDD	0.1739	0.1905	0.0000	0.0000	0.0000	0.8889	0.2609	0.9296	0.0000	0.2157	0.0000	0.0000	0.0000	0.5167	0.0000	0.0606	0.4444
LOF	0.3636	0.0561	0.2222	0.2564	0.3333	0.4000	0.3241	0.1500	0.0543	0.0873	0.2500	0.2509	0.0821	0.1307	0.0000	0.3200	0.5455
LUNAR	0.3478	0.2231	0.1333	0.3529	0.4314	0.8889	0.3054	0.1552	0.0571	0.0769	0.3103	0.1567	0.1416	0.1847	0.0690	0.5789	0.3750
MCD	0.4211	0.4330	0.0000	0.1111	0.1519	0.3333	0.2171	0.5487	0.0000	0.1040	0.2523	0.4730	0.2553	0.8157	0.0000	0.0500	0.3810
OCSVM	0.3636	0.4632	0.1818	0.2564	0.1839	0.4000	0.4116	0.4960	0.0089	0.2480	0.3186	0.4747	0.2697	0.8241	0.0000	0.1695	0.5000
PCA	0.3636	0.5895	0.1818	0.1053	0.0930	0.4444	0.3981	0.4769	0.0000	0.2335	0.3448	0.4581	0.2759	0.8174	0.0000	0.1429	0.5455
Proposed	0.3871	0.3939	0.3333	0.5946	0.2264	0.0000	0.0103	0.8333	0.0000	0.0566	0.5614	0.5111	0.7119	0.7811	0.0000	0.2727	0.5556
Sampling	0.2927	0.4186	0.1176	0.5000	0.3684	0.6154	0.3491	0.3594	0.0000	0.2385	0.3214	0.3017	0.1257	0.6247	0.0000	0.3188	0.4444

fectively handles datasets with varying characteristics. Its particularly strong performance on larger datasets with a high prevalence of outliers reinforces its suitability for real-world outlier detection scenarios.

Post hoc analysis using Holm adjustment (Table 4.6) highlights the superiority of the proposed method over several lower ranking models, such as LMDD, KPCA, and ABOD, which consistently perform poorly on datasets where the proposed method excels.

However, the analysis also does not show statistically significant differences between the proposed method and higher-ranking models like IForest, OCSVM, and Autoencoder. This indicates that these models are closely matched in terms of F1 performance, underscoring the competitive position of the proposed method among state-of-the-art approaches.

Table 4.6: F1 Score: Ranks and post hoc analysis of different outlier detection models

Ranking using the Friedman test

Model	Ranking
IForest	9.35294
OCSVM	9.67647
AutoEncoder	9.91176
Proposed	10.47059
PCA	10.79412
ECOD	10.94118
DeepSVDD	11.02941
HBOS	11.20588
KNN	11.52941
GMM	11.55882
COPOD	11.67647
CBLOF	11.79412
INNE	12.23529
Average KNN	12.73529
Median KNN	12.79412
KDE	13.05882
Sampling	13.35294
LUNAR	13.47059
LOF	14.44118
MCD	14.55882
Deep IForest	14.79412
ABOD	15.05882
KPCA	15.58824
LMDD	17.97059

Post hoc analysis using Holm's adjustment

Comparison	adj. p-value	p-value	z-value
Proposed vs LMDD	0.00000	0.00000	7.50000
Proposed vs KPCA	0.00001	0.00000	5.11765
Proposed vs ABOD	0.00009	0.00000	4.58824
Proposed vs Deep IForest	0.00031	0.00002	4.32353
Proposed vs MCD	0.00106	0.00006	4.02941
Proposed vs LOF	0.00165	0.00009	3.91176
Proposed vs DeepSVDD	0.02804	0.00165	3.14706
Proposed vs LUNAR	0.05231	0.00327	2.94118
Proposed vs Sampling	0.05921	0.00395	2.88235
Proposed vs KDE	0.15996	0.01143	2.52941
Proposed vs Median KNN	0.30590	0.02353	2.26471
Proposed vs Average KNN	0.32871	0.02739	2.20588
Proposed vs INNE	0.96833	0.08803	1.70588
Proposed vs CBLOF	1.00000	0.22786	1.20588
Proposed vs IForest	1.00000	0.26372	1.11765
Proposed vs COPOD	1.00000	0.27649	1.08824
Proposed vs GMM	1.00000	0.30329	1.02941
Proposed vs KNN	1.00000	0.31731	1.00000
Proposed vs OCSVM	1.00000	0.39369	0.85294
Proposed vs HBOS	1.00000	0.49874	0.67647
Proposed vs AutoEncoder	1.00000	0.53681	0.61765
Proposed vs ECOD	1.00000	0.68051	0.41176
Proposed vs PCA	1.00000	0.74629	0.32353

4.4.4 MCC Results

This section evaluates the performance of the proposed method using the Matthews Correlation Coefficient (MCC) across various datasets. The analysis includes correlations between the characteristics of the dataset and the performance of the MCC, the raw MCC scores for each dataset and model (Table 4.7), the Friedman ranking (Table 4.8) and post hoc comparisons using Holm adjustment (Table 4.8).

4.4.4.1 Correlation of Dataset Characteristics with MCC Performance

The correlation analysis (Figure 4.8) highlights how the characteristics of the dataset influence the performance of the MCC. Similarly to the analysis of the F1 score, the number of samples exhibits the strongest positive correlation with MCC (0.4), indicating that the method performs better on larger datasets, such as musk and shuttle. The number of outliers also shows a positive correlation (0.34), which further supports the method's ability to handle datasets with a high prevalence of outliers.

Dimensionality demonstrates a moderate correlation (0.2), suggesting that although higher-dimensional datasets present challenges, the method still performs reasonably well. Interestingly, the outlier percentage has a near-zero correlation (-0.02), indicating that the proportion of outliers in a dataset has minimal impact on the performance of the MCC.

In general, trends confirm that the proposed method is particularly suited to datasets with larger sample sizes and a higher number of outliers, aligning with its strong MCC performance on datasets like musk and shuttle (Table 4.7).

4.4.4.2 Raw MCC Results and Dataset-Specific Insights

A detailed analysis of the raw MCC results (Table 4.7) highlights the strengths and limitations of the proposed method.

- **Strong Performers**: The method achieves its highest MCC scores in datasets such as musk (MCC: 0.8354), shuttle (MCC: 0.7669), and satimage-2 (MCC: 0.7201). These datasets are characterized by large sample sizes and a sub-

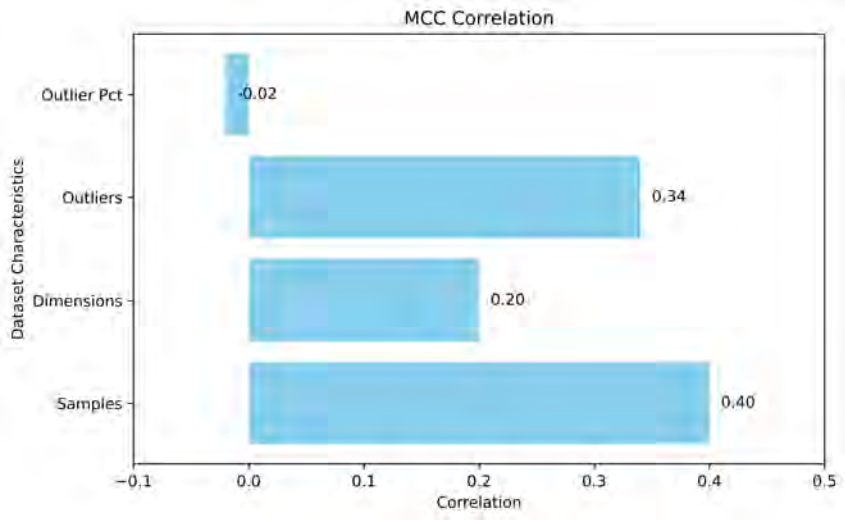

Figure 4.8: Correlations between the MCC score and different dataset characteristics.

stantial number of outliers, enabling the model to effectively learn and differentiate patterns for outlier detection. The strong MCC performance in these datasets aligns with the observed correlation trends, where the number of samples and outliers positively influences the performance.

- **Weak Performers**: The method underperforms on datasets such as lympho (MCC: -0.0325), vertebral (MCC: -0.1705), and optdigits (MCC: -0.0226). These datasets often feature small sample sizes or complex distributions that are challenging to model. For instance, lympho, with only 148 samples, highlights the difficulty of generalizing from limited data, as evidenced by its low MCC score.

4.4.4.3 Friedman Ranking and Holm's post hoc analysis

The Friedman ranking (Table 4.8) positions the proposed method 6th among 24 models, reflecting its competitive performance across diverse datasets. Although models such as IForest and OCSVM outperform it, the proposed method surpasses others like PCA and HBOS, showcasing its robustness and adaptability to varying dataset characteristics. This ranking highlights the method's general applicability and reliability in outlier detection tasks. Post hoc analysis using Holm adjustment (Table 4.8) reinforces the competitive standing of the proposed method. Significant performance differences were observed between the proposed method and

Table 4.7: MCC: Raw results per dataset and model

Model / Dataset	arrhythmia	cardio	glass	ionosphere	letter	lympho	mnist	musk	optdigits	pendigits	pima	satellite	satimage-2	shuttle	vertebral	vowels	wbc
ABOD	0.2009	0.1128	0.0989	0.2882	0.4110	0.3116	0.2867	-0.0337	-0.0007	0.0101	0.2417	0.1029	0.2407	0.1189	-0.1501	0.4408	0.4653
Average KNN	0.1793	0.2071	0.0237	0.3984	0.4366	0.6717	0.3410	0.1838	-0.0544	0.0614	0.1925	0.2704	0.2329	0.1531	-0.0210	0.4861	0.4114
AutoEncoder	0.2304	0.3057	-0.0563	0.5741	0.2612	0.2669	0.3069	-0.0271	0.5171	0.1444	0.1907	0.1793	0.4236	0.8962	0.0331	0.4685	0.3082
COPOD	0.3118	0.4234	0.0860	0.2936	-0.0027	0.7277	0.1404	0.4764	-0.0320	0.1939	0.2924	0.3644	0.2940	0.8350	-0.1540	-0.0285	0.4374
CBLOF	0.2887	0.4073	0.0989	0.2882	0.1950	0.3792	0.3624	0.5671	-0.0277	0.0698	0.1986	0.3046	0.3471	0.2312	-0.1803	0.3510	0.5026
Deep IForest	0.2009	0.3584	0.0000	0.2747	0.1531	0.5666	0.3699	0.4932	0.0100	0.2790	-0.0479	0.0479	0.4608	0.1661	-0.0555	0.0000	-0.0241
DeepSVDD	0.2516	0.4168	-0.0456	0.3714	0.0462	0.6100	0.1912	-0.0503	0.4622	0.2221	0.3097	0.3668	0.8091	0.5500	-0.1077	0.0402	0.1033
ECOD	0.3397	0.3948	0.0514	0.3568	0.0275	0.7277	0.0971	0.5078	-0.0237	0.2603	0.2104	0.3482	0.2643	0.8275	0.1780	0.0887	0.4374
GMM	0.0470	0.3547	0.0369	0.5583	0.2640	0.4205	0.3304	0.4156	-0.0530	0.0904	0.2627	0.2600	0.3103	0.7918	-0.0322	0.4027	0.4830
HBOS	0.5005	0.3826	0.0552	-0.2797	0.1112	0.3792	-0.0219	0.4940	0.2975	0.2835	0.3053	0.4512	0.3967	0.8622	-0.1030	0.1285	0.4634
IForest	0.5005	0.4383	0.0989	0.1815	0.0401	0.4638	0.1910	0.5815	-0.0277	0.3011	0.2790	0.4513	0.3769	0.8288	-0.1501	0.2031	0.5250
INNE	0.0880	0.3286	0.0114	0.4411	0.2349	0.6717	0.2935	0.5207	-0.0539	0.2415	0.2073	0.2865	0.3113	0.8310	-0.0428	0.3506	0.3903
KDE	0.0000	0.1542	0.2084	0.6868	0.1141	0.0000	0.0091	-0.4200	-0.1350	0.1644	0.2361	0.2902	0.2679	0.8345	-0.0529	0.4018	0.0565
KNN	0.2887	0.1931	0.1176	0.2484	0.3134	0.4638	0.3488	0.2335	-0.0252	0.1058	0.2657	0.3214	0.3024	0.1436	-0.1803	0.4482	0.5026
KPCA	0.0000	0.0316	0.0000	0.0000	0.0158	0.0000	0.0000	0.0000	-0.3652	0.0000	0.0000	0.0000	0.0000	0.3839	0.0000	-0.3271	-0.3678
LMDD	0.2203	0.2464	-0.0361	0.0000	0.0000	0.8835	0.2263	0.9293	-0.0282	0.1970	0.0000	0.0000	0.0000	0.5663	-0.0389	0.0242	0.5225
LOF	0.2887	-0.0423	0.1663	0.3238	0.2900	0.4638	0.2558	0.1222	0.0103	0.0675	0.1136	0.1465	0.0934	0.0499	-0.1394	0.3760	0.5250
LUNAR	0.2205	0.1103	0.0369	0.3760	0.4004	0.8835	0.2564	0.1228	0.0074	0.0519	0.1820	0.1427	0.1706	0.1098	-0.1159	0.5607	0.3316
MCD	0.3280	0.3819	0.0000	0.2018	0.0853	0.3116	0.1369	0.5969	-0.0607	0.0883	0.1259	0.4782	0.3666	0.8108	-0.1605	0.0190	0.3336
Median KNN	0.1793	0.1916	0.0369	0.3760	0.4043	0.6247	0.3613	0.2634	-0.0315	0.0514	0.1926	0.2697	0.2352	0.1452	-0.0210	0.4767	0.3885
OCSVM	0.2887	0.4151	0.1176	0.3238	0.1206	0.4638	0.3541	0.5535	-0.0504	0.2815	0.2117	0.4851	0.3795	0.8204	-0.1501	0.1881	0.4822
PCA	0.2887	0.5540	0.1176	0.0760	0.0150	0.5025	0.3392	0.5377	-0.0617	0.2649	0.2263	0.4507	0.3850	0.8130	-0.1160	0.1470	0.5250
Proposed	0.3337	0.3790	0.3251	0.3823	0.1825	-0.0325	0.0687	0.8354	-0.0226	0.0319	0.3037	0.3112	0.7201	0.7669	-0.1705	0.2552	0.5218
Sampling	0.1673	0.3496	0.0114	0.4815	0.3358	0.6247	0.2839	0.3763	-0.0670	0.2824	0.2358	0.2141	0.1876	0.5985	-0.1240	0.3192	0.4114

lower-ranking models such as KPCA and LMDD. However, no statistically significant differences emerged in comparisons with top-ranking models like IForest and OCSVM, indicating comparable MCC performance and positioning the proposed method among the best-performing models.

4.4.5 Precision Results

This section evaluates the Precision performance of the proposed method in various datasets, including an analysis of correlations with the characteristics of the dataset, the raw Precision scores (Table 4.9), the Friedman ranking (Table 4.10), and post hoc comparisons using Holm adjustment (Table 4.10).

4.4.5.1 Correlation of Dataset Characteristics with Precision Performance

Figure 4.9 illustrates the correlations between precision and key characteristics of the dataset. The number of outliers exhibits the strongest correlation (0.43), followed by the number of samples (0.37). Dimensionality (0.30) and the percent-

Table 4.8: MCC: Ranks and post hoc analysis of different outlier detection models

Ranking using the Friedman test

Model	Ranking
IForest	9.02941
OCSVM	9.35294
HBOS	9.94118
PCA	10.17647
ECOD	10.41176
Proposed	10.47059
AutoEncoder	11.47059
GMM	11.47059
INNE	11.47059
CBLOF	11.50000
KNN	11.67647
COPOD	11.76471
DeepSVDD	11.82353
Average KNN	12.64706
Median KNN	12.76471
Sampling	12.91176
Deep IForest	13.61765
LOF	13.94118
LUNAR	13.94118
KDE	14.35294
ABOD	14.55882
MCD	14.61765
LMDD	14.91176
KPCA	21.17647

Post hoc analysis using Holm's adjustment

Comparison	adj. p-values	p-values	z-values
Proposed vs KPCA	0.00000	0.00000	10.70588
Proposed vs LMDD	0.00020	0.00001	4.44118
Proposed vs MCD	0.00071	0.00003	4.14706
Proposed vs ABOD	0.00087	0.00004	4.08824
Proposed vs KDE	0.00197	0.00010	3.88235
Proposed vs LOF	0.00935	0.00052	3.47059
Proposed vs LUNAR	0.00935	0.00052	3.47059
Proposed vs Deep IForest	0.02639	0.00165	3.14706
Proposed vs Sampling	0.21959	0.01464	2.44118
Proposed vs Median KNN	0.30497	0.02178	2.29412
Proposed vs Average KNN	0.38376	0.02952	2.17647
Proposed vs IForest	1.00000	0.14953	1.44118
Proposed vs DeepSVDD	1.00000	0.17607	1.35294
Proposed vs COPOD	1.00000	0.19562	1.29412
Proposed vs KNN	1.00000	0.22786	1.20588
Proposed vs OCSVM	1.00000	0.26372	1.11765
Proposed vs CBLOF	1.00000	0.30329	1.02941
Proposed vs AutoEncoder	1.00000	0.31731	1.00000
Proposed vs GMM	1.00000	0.31731	1.00000
Proposed vs INNE	1.00000	0.31731	1.00000
Proposed vs HBOS	1.00000	0.59652	0.52941
Proposed vs PCA	1.00000	0.76867	0.29412
Proposed vs ECOD	1.00000	0.95309	0.05882

age of outliers (0.22) also show positive correlations, indicating that the proposed method generally performs better in datasets with more outliers and larger sample sizes. Although dimensionality shows a moderate impact, some higher-dimensional datasets may still pose challenges.

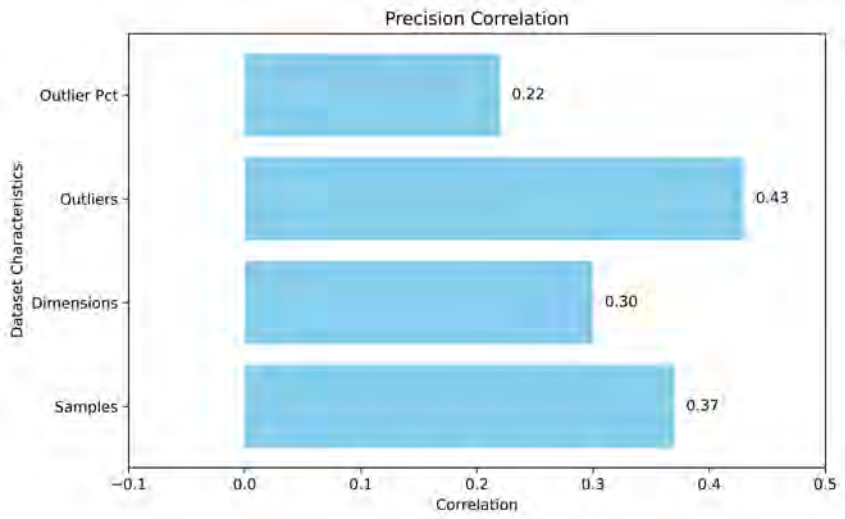

Figure 4.9: Correlations between Precision and various dataset characteristics, including outlier percentage, number of outliers, dimensionality, and number of samples.

These results suggest that the proposed method is well suited for datasets with a substantial number of outliers and larger sample sizes, demonstrating consistent adaptability to datasets with varying dimensions.

4.4.5.2 Raw Precision Results and Dataset-Specific Insights

An analysis of raw precision scores across datasets (Table 4.9) highlights both the strengths and limitations of the proposed method:

- **Strong Performers**: The proposed method achieves high Precision scores on datasets such as musk (Precision: 0.7317), shuttle (Precision: 0.8312), and satimage-2 (Precision: 0.6000). These datasets, characterized by larger sample sizes and substantial outlier counts, enable the model to effectively learn and detect patterns indicative of outliers.
- **Weak Performers**: The method struggles on datasets such as vertebral (Precision: 0.0000), pendigits (Precision: 0.0526) and lympho (Precision: 0.0000). These datasets typically feature small sample sizes and lower outlier

counts, limiting the model's ability to generalize effectively.

Table 4.9: Precision: Raw results per dataset and model

Model / Dataset	arrhythmia	cardio	glass	ionosphere	letter	lympho	mnist	musk	optdigits	pendigits	pima	satellite	satimage-2	shuttle	vertebral	vowels	wbc
ABOD	0.3571	0.1639	0.1250	1.0000	0.3542	0.2500	0.3416	0.0133	0.0291	0.0279	0.7273	0.4574	0.1039	0.1644	0.0000	0.2128	0.4167
AutoEncoder	0.2857	0.3418	0.0000	0.6944	0.3077	0.2000	0.3796	0.0000	0.6190	0.2414	0.5278	0.4415	0.3913	0.9025	0.1250	0.4286	0.3000
Average KNN	0.3000	0.2609	0.0909	1.0000	0.3276	0.5000	0.3675	0.1296	0.0056	0.0500	0.6522	0.6837	0.0811	0.1880	0.0833	0.2833	0.3636
CBLOF	0.4615	0.4444	0.1250	1.0000	0.1833	0.1818	0.3937	0.3444	0.0165	0.0545	0.6207	0.7459	0.1326	0.2539	0.0000	0.1778	0.4000
COPOD	0.4211	0.4516	0.1429	0.8182	0.0545	0.5714	0.2000	0.3210	0.0144	0.1095	0.7692	0.8280	0.1076	0.7270	0.0000	0.0238	0.4000
Deep IForest	0.3571	0.6087	0.0000	1.0000	0.4000	0.5000	0.3837	0.6087	0.0405	0.4286	0.2500	1.0000	0.2209	0.8605	0.0000	0.0000	0.0000
DeepSVDD	0.3077	0.4912	0.0000	0.5526	0.0968	0.4000	0.2814	0.0423	0.5517	0.2759	0.6053	0.5836	0.8333	0.5729	0.0000	0.0556	0.1667
ECOD	0.4286	0.4219	0.1111	0.9091	0.0741	0.5714	0.1639	0.3462	0.0197	0.1393	0.7222	0.8156	0.0947	0.7182	0.2000	0.0952	0.4000
GMM	0.1520	0.3768	0.1000	1.0000	0.2308	0.2500	0.3494	0.2025	0.0058	0.0649	0.7143	0.6811	0.1061	0.6888	0.0769	0.2344	0.3846
HBOS	0.6923	0.4222	0.0909	0.0500	0.1373	0.1818	0.0682	0.2696	0.1758	0.1517	0.7419	0.9471	0.1690	0.7893	0.0000	0.0741	0.3529
IForest	0.6923	0.4423	0.1250	0.6667	0.0862	0.2500	0.2445	0.3605	0.0165	0.1626	0.7097	0.9381	0.1538	0.7175	0.0000	0.1136	0.4286
INNE	0.2069	0.3521	0.0833	1.0000	0.2115	0.5000	0.3277	0.2973	0.0057	0.1280	0.6667	0.7341	0.1067	0.7257	0.0714	0.2308	0.2778
KDE	0.1471	0.1677	0.1667	0.7451	0.0685	0.0889	0.0856	0.0296	0.0298	0.0905	0.6471	0.6898	0.0819	0.7277	0.0667	0.2073	0.0690
KNN	0.4615	0.2321	0.1429	1.0000	0.2955	0.2500	0.3980	0.1667	0.0173	0.0667	0.6875	0.7632	0.1242	0.1826	0.0000	0.2432	0.4000
KPCA	0.1471	0.0991	0.0769	0.3962	0.0565	0.0889	0.0855	0.0359	0.0282	0.0223	0.3723	0.3112	0.0109	0.3231	0.0972	0.0368	0.0531
LMDD	0.6667	0.6667	0.0000	0.0000	0.0000	0.8000	0.3750	0.8684	0.0000	0.1964	0.0000	0.0000	0.0000	0.9562	0.0000	0.0667	1.0000
LOF	0.4615	0.0469	0.2000	1.0000	0.2653	0.2500	0.2941	0.1011	0.0343	0.0531	0.5185	0.5155	0.0468	0.1087	0.0000	0.2000	0.4286
LUNAR	0.3077	0.1421	0.1000	1.0000	0.4583	0.8000	0.3669	0.1084	0.0361	0.0595	0.6000	0.6136	0.0851	0.1141	0.0455	0.5500	0.3333
MCD	0.4444	0.3889	0.0000	1.0000	0.1200	0.2500	0.1967	0.3780	0.0000	0.0647	0.5385	0.9792	0.1463	0.7059	0.0000	0.0333	0.3077
Median KNN	0.3000	0.2500	0.1000	1.0000	0.3051	0.4444	0.3808	0.1731	0.0169	0.0457	0.6667	0.6915	0.0824	0.1813	0.0833	0.2742	0.3333
OCSVM	0.4615	0.4231	0.1429	1.0000	0.1379	0.2500	0.3881	0.3298	0.0056	0.1542	0.6429	0.9947	0.1558	0.7119	0.0000	0.1020	0.3750
PCA	0.4615	0.5385	0.1429	0.5000	0.0702	0.2857	0.3761	0.3131	0.0000	0.1442	0.6452	0.9337	0.1600	0.7059	0.0000	0.0870	0.4000
Proposed	0.5455	0.5652	0.5000	0.5500	0.2500	0.0000	1.0000	0.7317	0.0000	0.0526	0.5581	0.5473	0.6000	0.8312	0.0000	0.2500	0.5000
Sampling	0.2857	0.3600	0.0833	1.0000	0.2857	0.4444	0.3231	0.2421	0.0000	0.1449	0.6923	0.6020	0.0698	0.5487	0.0000	0.2157	0.3636

4.4.5.3 Friedman Ranking and Holm's post hoc analysis

According to Friedman's ranking (Table 4.10), the proposed method is third among 24 models in terms of precision. This result highlights its strong competitive edge, outperforming established models such as HBOS, PCA, and AutoEncoder. The method's superior performance on datasets with larger sample sizes and higher outlier counts underscores its effectiveness and positions it as one of the leading models for precision-focused outlier detection tasks. The post hoc analysis conducted using the Holm adjustment method (Table 4.10) confirms the strong performance of the proposed method. In particular, significant differences were identified between the proposed method and lower-performing models such as KPCA, KDE, and MCD. In contrast, no significant differences emerged when comparing the proposed method with high-ranking models such as Deep IForest and OCSVM, indicating that these models perform similarly in terms of precision.

Table 4.10: Precision: Ranks and post hoc analysis of different outlier detection models

Ranking using the Friedman test

Model	Ranking
Deep IForest	9.50000
OCSVM	10.11765
Proposed	10.23529
IForest	10.29412
ECOD	10.52941
KNN	10.76471
PCA	11.35294
COPOD	11.50000
CBLOF	11.55882
HBOS	11.61765
AutoEncoder	11.97059
INNE	12.23529
GMM	12.35294
LUNAR	12.38235
Median KNN	12.38235
Average KNN	12.85294
LMDD	13.02941
ABOD	13.50000
Sampling	13.70588
DeepSVDD	13.97059
LOF	14.35294
MCD	14.38235
KDE	15.35294
KPCA	20.05882

Post hoc analysis using Holm's adjustment

Comparison	adj. p-values	p-values	z-values
Proposed vs KPCA	0.00000	0.00000	9.82353
Proposed vs KDE	0.00001	0.00000	5.11765
Proposed vs MCD	0.00071	0.00003	4.14706
Proposed vs LOF	0.00077	0.00004	4.11765
Proposed vs DeepSVDD	0.00356	0.00019	3.73529
Proposed vs Sampling	0.00935	0.00052	3.47059
Proposed vs ABOD	0.01863	0.00110	3.26471
Proposed vs LMDD	0.08327	0.00520	2.79412
Proposed vs Average KNN	0.13281	0.00885	2.61765
Proposed vs LUNAR	0.44504	0.03179	2.14706
Proposed vs Median KNN	0.44504	0.03179	2.14706
Proposed vs GMM	0.44504	0.03420	2.11765
Proposed vs INNE	0.50050	0.04550	2.00000
Proposed vs AutoEncoder	0.82689	0.08269	1.73529
Proposed vs HBOS	1.00000	0.16686	1.38235
Proposed vs CBLOF	1.00000	0.18566	1.32353
Proposed vs COPOD	1.00000	0.20598	1.26471
Proposed vs PCA	1.00000	0.26372	1.11765
Proposed vs Deep IForest	1.00000	0.46216	0.73529
Proposed vs KNN	1.00000	0.59652	0.52941
Proposed vs ECOD	1.00000	0.76867	0.29412
Proposed vs OCSVM	1.00000	0.90635	0.11765
Proposed vs IForest	1.00000	0.95309	0.05882

4.4.6 Recall Results

This section examines the performance of the proposed method in terms of recall in various datasets. Although the method performed exceptionally well in terms of precision, the inherent trade-off between precision and recall leads to some limitations in recall. The analysis includes correlations between the characteristics of the dataset and the recall, raw recall scores for each dataset and model (Table 4.11), Friedman's ranking (Table 4.12) and post hoc comparisons using Holm's adjustment (Table 4.12).

4.4.6.1 Correlation of Dataset Characteristics with Recall Performance

Figure 4.10 shows the correlation between the recall and key characteristics of the dataset. The strongest correlations are with the number of outliers and the number of samples (both at 0.27), followed by the outlier percentage (0.20). The weakest correlation is with dimensionality (0.11), indicating a minimal impact of the number of features on the recall.

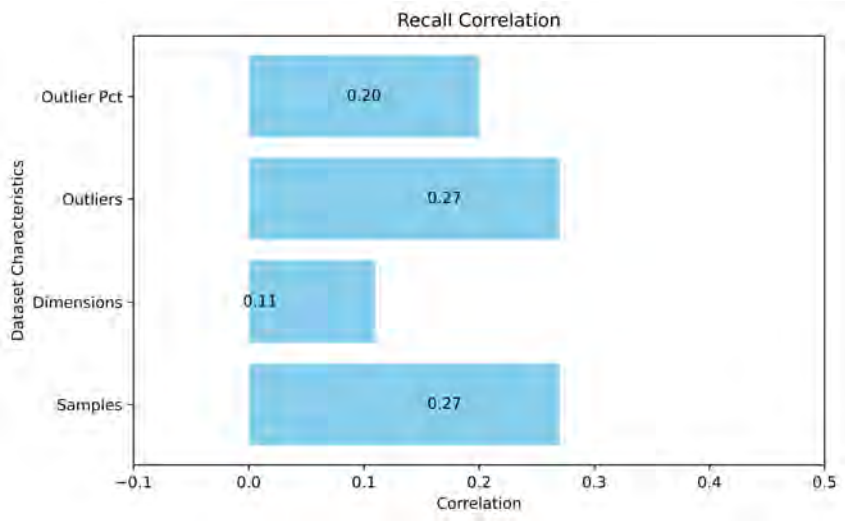

Figure 4.10: Correlations between Recall and dataset characteristics.

This analysis highlights the strength of the method in datasets with larger sample sizes and more outliers, but also underscores the challenges in balancing high Precision with Recall.

4.4.6.2 Raw Recall Results and Dataset-Specific Insights

The raw recall results (Table 4.11) demonstrate the trade-off between the high precision of the method and its recall performance:

- **Strong Performers**: High Recall is achieved on datasets such as musk (0.9677), shuttle (0.7367), and satimage-2 (0.8750). These datasets have larger sample sizes and a larger number of outliers, allowing the model to generalize effectively.
- **Weak Performers**: The method struggles with datasets like mnist (0.0052), vertebral (0.0000), and pendigits (0.0612), likely due to small sample sizes and the precision optimized nature of the model, which makes it less effective in capturing all relevant outliers.

Table 4.11: Recall: Raw results per dataset and model

Model / Dataset	arrhythmia	cardio	glass	ionosphere	letter	lympho	mnist	musk	optdigits	pendigits	pima	satellite	satimage-2	shuttle	vertebral	vowels	wbc
ABOD	0.2500	0.2326	0.2500	0.1176	0.5862	0.5000	0.3557	0.0323	0.1087	0.1429	0.1882	0.1426	0.6667	0.2263	0.0000	1.0000	0.6250
AutoEncoder	0.4444	0.4500	0.0000	0.7353	0.2963	0.5000	0.3677	0.0000	0.4444	0.1724	0.4222	0.4723	0.4737	0.9043	0.1429	0.5455	0.4286
Average KNN	0.3000	0.3333	0.2000	0.2381	0.7037	1.0000	0.4410	0.4242	0.0196	0.2174	0.1744	0.2230	0.7895	0.2691	0.1429	0.9444	0.5714
CBLOF	0.3000	0.4651	0.2500	0.1176	0.3793	1.0000	0.4485	1.0000	0.0652	0.2449	0.2118	0.2289	1.0000	0.3428	0.0000	0.8000	0.7500
COPOD	0.4000	0.5185	0.2000	0.2143	0.1111	1.0000	0.2462	0.7879	0.0392	0.4783	0.2326	0.2562	0.8947	0.9885	0.0000	0.0556	0.5714
Deep IForest	0.2500	0.2593	0.0000	0.1190	0.0741	0.7500	0.4821	0.4242	0.0588	0.1957	0.0233	0.0033	1.0000	0.0353	0.0000	0.0000	0.0000
DeepSVDD	0.4444	0.4667	0.0000	0.6176	0.1111	1.0000	0.2511	0.0847	0.5556	0.2414	0.5111	0.5531	0.7895	0.5908	0.0000	0.0909	0.1429
ECOD	0.4500	0.5000	0.2000	0.2381	0.1481	1.0000	0.2051	0.8182	0.0588	0.6087	0.1512	0.2429	0.8421	0.9847	0.4286	0.2222	0.5714
GMM	0.9500	0.4815	0.2000	0.4286	0.4444	1.0000	0.4462	1.0000	0.0196	0.2609	0.2326	0.2097	1.0000	0.9485	0.1429	0.8333	0.7143
HBOS	0.4500	0.4419	0.2500	0.0294	0.2414	1.0000	0.0928	1.0000	0.6304	0.6531	0.2706	0.2968	1.0000	0.9650	0.0000	0.4000	0.7500
IForest	0.4500	0.5349	0.2500	0.1176	0.1724	1.0000	0.2887	1.0000	0.0652	0.6735	0.2588	0.3018	1.0000	0.9886	0.0000	0.5000	0.7500
INNE	0.3000	0.4630	0.2000	0.2857	0.4074	1.0000	0.3949	1.0000	0.0196	0.5870	0.1860	0.2113	1.0000	0.9819	0.1429	0.6667	0.7143
KDE	1.0000	0.5185	0.6000	0.9048	1.0000	1.0000	1.0000	0.8182	0.9020	0.4565	0.2558	0.2479	1.0000	0.9866	0.1429	0.9444	0.8571
KNN	0.3000	0.3023	0.2500	0.0882	0.4483	1.0000	0.4124	0.4516	0.0652	0.3469	0.2588	0.2405	0.8333	0.2557	0.0000	0.9000	0.7500
KPCA	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8627	1.0000	1.0000	1.0000	1.0000	0.6040	1.0000	0.8889	0.8571
LMDD	0.1000	0.1111	0.0000	0.0000	0.0000	1.0000	0.2000	1.0000	0.0000	0.2391	0.0000	0.0000	0.0000	0.3540	0.0000	0.0556	0.2857
LOF	0.3000	0.0698	0.2500	0.1471	0.4483	1.0000	0.3608	0.2903	0.1304	0.2449	0.1647	0.1658	0.3333	0.1638	0.0000	0.8000	0.7500
LUNAR	0.4000	0.5185	0.2000	0.2143	0.4074	1.0000	0.2615	0.2727	0.1373	0.1087	0.2093	0.0899	0.4211	0.4838	0.1429	0.6111	0.4286
MCD	0.4000	0.4884	0.0000	0.0588	0.2069	0.5000	0.2423	1.0000	0.0000	0.2653	0.1647	0.3118	1.0000	0.9659	0.0000	0.1000	0.5000
Median KNN	0.3000	0.3148	0.2000	0.2143	0.6667	1.0000	0.4667	0.5455	0.0588	0.1957	0.1628	0.2163	0.7895	0.2624	0.1429	0.9444	0.5714
OCSVM	0.3000	0.5116	0.2500	0.1471	0.2759	1.0000	0.4381	1.0000	0.0217	0.6327	0.2118	0.3118	1.0000	0.9782	0.0000	0.5000	0.7500
PCA	0.3000	0.6512	0.2500	0.0588	0.1379	1.0000	0.4227	1.0000	0.0000	0.6122	0.2353	0.3035	1.0000	0.9706	0.0000	0.4000	0.7500
Proposed	0.3000	0.3023	0.2500	0.6471	0.2069	0.0000	0.0052	0.9677	0.0000	0.0612	0.5647	0.4793	0.8750	0.7367	0.0000	0.3000	0.6250
Sampling	0.3000	0.5000	0.2000	0.3333	0.5185	1.0000	0.3795	0.6970	0.0000	0.6739	0.2093	0.2013	0.6316	0.7252	0.0000	0.6111	0.5714

4.4.6.3 Friedman Ranking and Holm's post hoc analysis

According to Friedman rank (Table 4.12), the proposed method ranks 19th out of 24 models in terms of recall. This outcome reflects the inherent tradeoff between Pre-

cision and Recall, with the method's focus on optimizing Precision leading to lower performance in Recall. Models such as KPCA and KDE achieve higher Recall but often struggle with Precision, highlighting the difficulty of balancing these two metrics effectively. Post hoc analysis using Holm's adjustment method (Table 4.12) highlights significant differences between the proposed method and Recall-focused models like KPCA and KDE. In contrast, the differences with models such as ECOD and KNN are less pronounced, indicating that the proposed method remains competitive in scenarios where both Precision and Recall must be balanced.

4.4.7 Discussion

This section provides an interpretation of the results and a detailed discussion of the strengths and limitations of the proposed method.

4.4.7.1 Overview

The proposed method demonstrates competitive performance on various outlier detection metrics, excelling particularly in precision. Its ability to accurately identify true positives while minimizing false positives makes it ideal for applications where reducing false alarms is critical. Although it does not achieve the top ranking on the F1 score or the MCC, its performance is consistent and comparable to leading models like IForest and OCSVM, with no statistically significant differences observed. This reliability across datasets highlights the robustness of the approach.

A key strength of the method is its interpretability. Using decision trees produces clear and comprehensible decision rules, enabling practitioners to understand the factors that influence outlier detection. Additionally, the invertibility of the normalizing flow model provides additional insight into data transformations, enhancing transparency. These features make the method particularly valuable in domains where decision-making processes must be transparent and justifiable.

However, the method faces challenges with Recall, which struggle to capture all relevant outliers in some datasets. This tradeoff between Precision and Recall is most evident in scenarios where minimizing false positives takes precedence, potentially at the expense of missing some outliers. Future work focusing on optimization

Table 4.12: Recall: Ranks and post hoc analysis of different outlier detection models

Ranking using the Friedman test

Model	Ranking
KPCA	3.32353
KDE	5.02941
IForest	9.08824
GMM	9.70588
OCSVM	10.05882
HBOS	10.58824
PCA	11.05882
INNE	11.08824
CBLOF	11.38235
ECOD	12.35294
KNN	12.47059
COPOD	13.00000
AutoEncoder	13.23529
Sampling	13.26471
Average KNN	13.35294
Median KNN	13.38235
DeepSVDD	13.88235
LUNAR	14.26471
Proposed	14.47059
LOF	14.82353
MCD	15.08824
ABOD	16.05882
Deep IForest	18.79412
LMDD	20.23529

Post hoc analysis using Holm's adjustment

Comparison	adj. p-values	p-values	z-values
Decision Tree vs KDE	0.00000	0.00000	9.44118
Decision Tree vs KPCA	0.00000	0.00000	11.14706
Decision Tree vs LMDD	0.00000	0.00000	5.76471
Decision Tree vs IForest	0.00000	0.00000	5.38235
Decision Tree vs GMM	0.00004	0.00000	4.76471
Decision Tree vs OCSVM	0.00018	0.00001	4.41176
Decision Tree vs Deep IForest	0.00026	0.00002	4.32353
Decision Tree vs HBOS	0.00166	0.00010	3.88235
Decision Tree vs PCA	0.00968	0.00065	3.41176
Decision Tree vs INNE	0.01006	0.00072	3.38235
Decision Tree vs CBLOF	0.02618	0.00201	3.08824
Decision Tree vs ECOD	0.41046	0.03420	2.11765
Decision Tree vs KNN	0.50050	0.04550	2.00000
Decision Tree vs ABOD	1.00000	0.11223	1.58824
Decision Tree vs COPOD	1.00000	0.14140	1.47059
Decision Tree vs AutoEncoder	1.00000	0.21672	1.23529
Decision Tree vs Sampling	1.00000	0.22786	1.20588
Decision Tree vs Average KNN	1.00000	0.26372	1.11765
Decision Tree vs Median KNN	1.00000	0.27649	1.08824
Decision Tree vs MCD	1.00000	0.53681	0.61765
Decision Tree vs DeepSVDD	1.00000	0.55637	0.58824
Decision Tree vs LOF	1.00000	0.72413	0.35294
Decision Tree vs LUNAR	1.00000	0.83688	0.20588

or fine-tuning could improve the balance between these metrics, allowing the model to better detect outliers without compromising precision.

In terms of computational efficiency, the method excels in fast inference, making it well-suited for real-time applications. However, computational complexity during training, particularly for large datasets, presents a limitation. Moreover, the current evaluation is restricted to tabular and image data, leaving its applicability to other data types, such as time series or graph data, unexplored.

In summary, the proposed method offers a robust and interpretable solution for outlier detection. Its strong performance in key metrics, rapid inference, and clear decision-making processes make it a competitive choice for applications requiring precision and interpretability. Although improvements in recall and expanded testing across diverse data types would enhance its applicability, the method is well suited for datasets with larger sample sizes and significant outlier proportions.

4.4.7.2 Strengths of the Proposed Method

The proposed method offers several distinct advantages:

- **Adaptability**: Using normalizing flows, the method effectively models diverse data distributions, enabling accurate outlier detection across datasets with varying complexities and characteristics.
- **Transparency**: The integration of decision trees provides interpretable decision rules, allowing practitioners to understand the factors driving the classification of inliers and outliers. This enhances trust and usability in real-world applications.
- **Robust Performance**: The method achieves competitive results across critical metrics such as Precision, Recall, F1, and MCC. Post hoc analyzes confirm that no competing models significantly outperform it, highlighting its reliability and robustness.
- **Efficiency**: With the fastest scoring times among all models evaluated, the method is well suited for applications requiring rapid or real-time anomaly detection.
- **Synthetic Data Generation**: The ability to generate synthetic datasets from

the learned distribution adds scalability. This feature is particularly useful in unbalanced datasets, enabling the model to improve detection performance by increasing the representation of minority classes.

4.4.7.3 Limitations

Although the proposed method demonstrates numerous strengths, it also has some limitations:

- **Challenges with Small Datasets**: The model struggles to generalize effectively on datasets with limited samples, such as Lympho and Vertebral. This leads to suboptimal results, highlighting the dependence of the model on sufficient data for robust learning.
- **Precision-Recall Tradeoff**: The method prioritizes precision at the expense of recall, which is evident in datasets with fewer outliers or significant class imbalance. Although this focus minimizes false positives, it limits the model's ability to capture all relevant outliers in certain scenarios.
- **High Computational Demand**: The training process for normalizing flows is computationally intensive, particularly on large datasets. This could be a concern in resource-limited environments or applications that require frequent model retraining.
- **Dependence on Synthetic Data Quality**: The performance of the decision tree is heavily influenced by the quality of the synthetic data generated by the normalizing flow. Although the method performed well on 17 datasets, datasets with lower-quality synthetic data may pose challenges.
- **Limited Testing Scope**: The approach has been evaluated only on tabular and image datasets. Other data types, such as time series or graph-structured data, remain untested, potentially limiting its applicability and generalizability to these domains.

4.4.7.4 Balancing Interpretability, Performance, and Real-World Applicability

Machine learning models often face a trade-off between interpretability and performance. Simple, interpretable models can sacrifice predictive accuracy, while complex, high-performing models can lack transparency. This balance is especially important in real-world applications, where the prioritization of specific metrics depends on the use case.

For our model, the primary consideration is its trade-off between interpretability and performance, particularly its 19th-place ranking in recall. Although this may be a limitation in certain contexts, the strong results of the model in precision, F1 score, and MCC demonstrate its overall robustness. Importantly, the statistical significance tests ($p = 0.05$) showed that the model did not perform worse than competing methods on any metric while significantly outperforming several baseline models. This suggests that despite lower recall, the model excels in applications where precision is critical, effectively minimizing false positives.

In domains like medical diagnostics, where false positives can lead to unnecessary treatments or fraud detection, where overflagging incurs financial and operational costs, precision and interpretability are essential. Here, our model's ability to reduce false positives and provide clear, explainable decision-making makes it a strong choice. Precision ensures fewer incorrect positive predictions, while interpretability helps domain experts understand and trust the model output.

In contrast, in scenarios such as monitoring disease outbreaks or search-and-rescue operations, where missing a positive event has severe consequences, recall becomes the priority. In these contexts, models designed to capture as many positives as possible, even at the cost of more false positives, may be preferable. Although our method excels in precision and interpretability, it may not be ideal for such recall-centric applications unless transparency is a higher priority than recall.

This Precision-Recall trade-off is not unique to our approach, but is observed across many machine learning models. For example, DeepIForest achieves top Precision rankings but struggles with Recall, whereas KDE and KPCA, which perform

poorly in Precision, lead in Recall rankings. These trends underscore that no single model excels in all metrics; the optimal choice depends on the specific needs and objectives of the application.

In summary, while our model's recall ranking is lower, its strengths in precision, F1 score, MCC, and interpretability make it highly suitable for applications where minimizing false positives and decision transparency are priorities. For tasks where recall is critical, alternative models may be better suited.

4.4.7.5 Dataset-Specific Performance Analysis

This section examines the datasets where the proposed model underperformed—specifically lympho, vertebral, and optdigits—and contrasts them with datasets that share similar characteristics but yielded better results. Across these three datasets, both precision and recall were 0, indicating that the model did not detect any outliers.

Lympho

The lympho dataset presents significant challenges due to its small sample size (148) and low outlier percentage (4.05%). The 18 features of the dataset probably exacerbate the problem by introducing complexity that increases the risk of overfitting given the limited number of instances. In contrast, the glass dataset, which has a similar sample size (214) and an outlier percentage (4. 21%) but fewer features (9), performs better. This suggests that a smaller feature set can help avoid overfitting in small datasets, enabling the model to better distinguish between normal and outlier instances.

Vertebral

The vertebral dataset also performed poorly, despite having a larger sample size (240) and a higher outlier percentage (12.5%). Its low dimensionality (6 features) probably limits the model's ability to capture meaningful patterns, resulting in both precision and recall scores of 0. Interestingly, this dataset posed challenges for most models, with several other algorithms, including HBOS, IForest, and PCA, also failing to detect outliers. By comparison, the glass dataset, with slightly higher dimensionality (9 features), performed better, reinforcing the idea that datasets with

very few features may not provide enough variability for effective outlier detection.

Optdigits

The optdigits dataset, with 5216 samples and 64 features, offers a richer feature space but suffers from an extremely low outlier percentage (2.88%). This class imbalance, combined with the high dimensionality, probably explains why the model failed to detect any outliers (precision and recall of 0). Other algorithms, such as MCD and PCA, also struggled on this dataset, indicating that its structure poses challenges for many outlier detection methods. By contrast, the pendigits dataset, with more samples (6870), fewer features (16), and a similarly low outlier percentage (2.27%), yielded much better results. This suggests that lower dimensionality allows the model to focus on relevant features, improving detection performance.

Key Insights from Comparison

The analysis reveals three critical factors affecting performance across datasets:

- **Class Imbalance**: Low outlier percentages, as seen in lympho and optdigits, create significant challenges to class imbalance. In contrast, datasets with higher outlier percentages, such as satellites, offer more balanced opportunities for learning and better performance.
- **Sample Size**: Larger sample sizes generally improve model performance, but the interplay between sample size and dimensionality is crucial. Datasets like pendigits and satellite, with balanced sample sizes and feature counts, enable better generalization without overfitting.
- **Dimensionality**: Very low dimensionality, as in vertebral, fails to provide sufficient variability for the model, while excessive dimensionality, as in optdigits, can introduce noise and hinder performance. Striking a balance in feature count is essential for effective learning.

The underperformance of the proposed model on datasets like lympho, vertebral, and optdigits reflects a combination of class imbalance, limited sample sizes, and the interaction between dimensionality and data distribution. These insights highlight the importance of considering the characteristics of the dataset when applying the model in practical scenarios, particularly for datasets with extreme im-

balances or challenging feature distributions.

4.4.7.6 Sensitivity Analysis of Threshold Choices

This section evaluates the impact of threshold selection on model performance using ROC curves for 17 datasets, including Arrhythmia, Cardio, Glass, Ionosphere, Letter, Lympho, MNIST, Musk, Optdigits, Pendigits, Pima, Satellite, Satimage-2, Shuttle, Vertebral, Vowels, and WBC. ROC curves illustrate the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR) at various thresholds, helping to identify optimal thresholds for balancing sensitivity and specificity. Figure 4.11 presents the ROC curves, highlighting dataset-specific behaviors and performance trends.

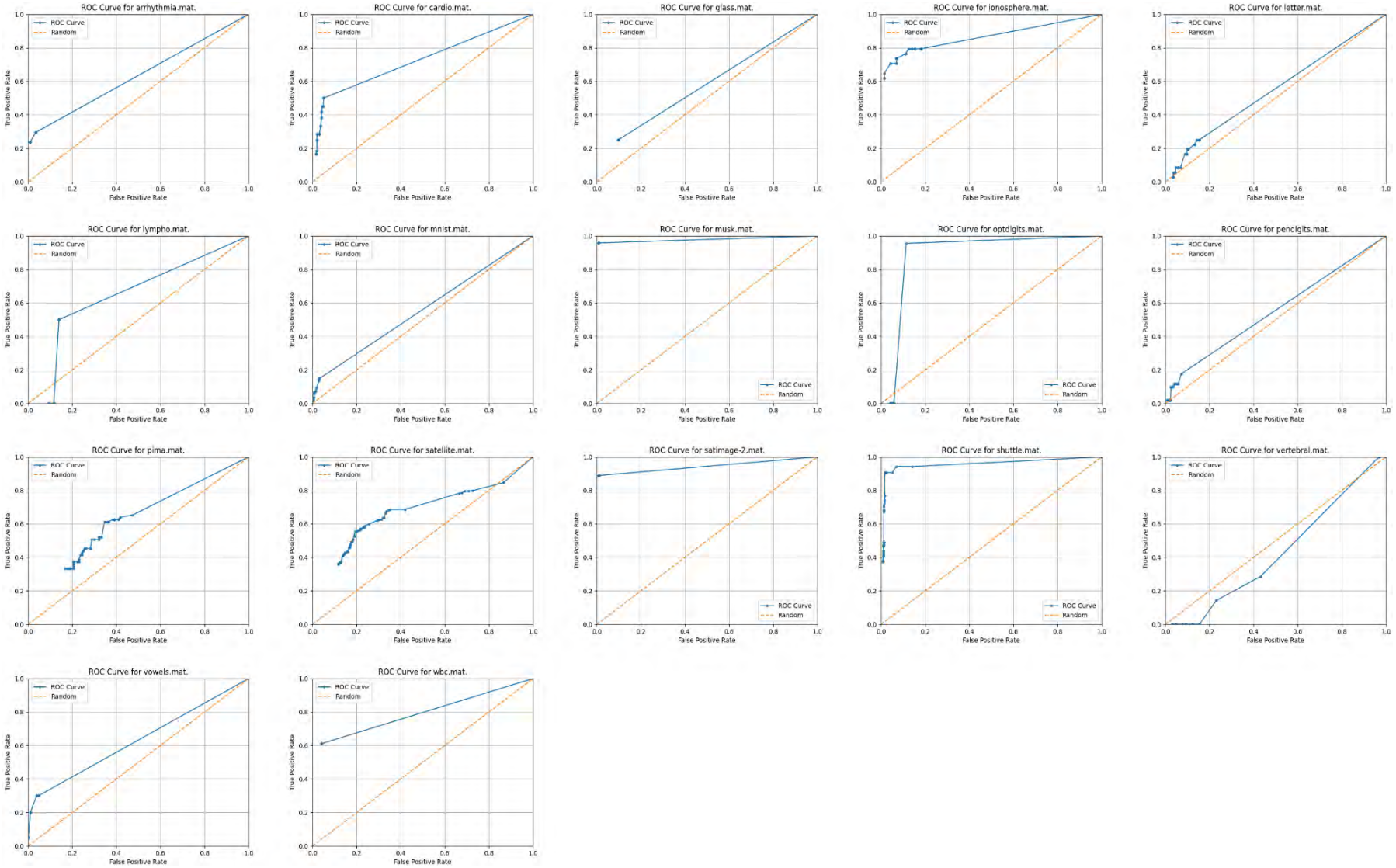

Figure 4.11: ROC Curves for 17 datasets, illustrating the trade-off between TPR and FPR. The diagonal line represents random classification. Curves closer to the top-left corner indicate better performance, while variations in curve shapes reflect dataset-specific sensitivities to threshold adjustments.

Dataset-Specific Observations

The analysis reveals three distinct patterns of sensitivity to threshold choices across the datasets:

- **Highly Sensitive Datasets:** - Datasets like Cardio, Ionosphere, Lympho, Satellite, and Vowels exhibit steep initial increases in their ROC curves. Small threshold changes at lower FPR lead to significant improvements in TPR, indicating high sensitivity to threshold adjustments. For these datasets, careful adjustment of thresholds is critical to balance detection accuracy and false positives.
- **Stable Datasets:** - Datasets such as Musk, Optdigits, Satimage-2, and Shuttle achieve near-perfect ROC curves with TPRs approaching 1 at very low FPRs. These datasets show minimal sensitivity to threshold changes, allowing greater flexibility in threshold selection without sacrificing performance.
- **Moderately Sensitive Datasets:** - Datasets including Arrhythmia, Glass, MNIST, Pima, Pendigits, Vertebral, and WBC display more linear ROC curves. Threshold adjustments result in proportional changes in TPR and FPR, making performance predictable and tuning thresholds less critical.

Illustrative Examples

Cardio and Ionosphere: These datasets show sharp performance improvements at low thresholds, emphasizing the importance of precise tuning for optimal results.

Musk and Optdigits: With nearly perfect ROC curves, these datasets demonstrate robust performance across all thresholds, making threshold selection less consequential.

Lympho and Vertebral: The steep early rises in these datasets indicate high sensitivity to threshold adjustments, but performance gains plateau quickly with further increases.

Overall Trends

The sensitivity analysis highlights the diverse behaviors of the datasets in response to threshold adjustments. Although highly sensitive datasets benefit significantly from precise threshold selection, stable datasets provide flexibility in tuning with-

out substantial performance degradation. This underscores the importance of tailoring threshold selection to dataset-specific characteristics for optimal detection performance.

In conclusion, datasets with steep ROC curves at low FPRs, such as Cardio and Satellite, require careful tuning, while datasets like Musk and Shuttle offer robust performance across thresholds. For most datasets, consistent performance trends simplify threshold optimization, ensuring reliable application in diverse scenarios.

4.4.7.7 Global Interpretability from Feature Importances

The feature importance derived from the decision tree model provides critical insight into the factors driving classification decisions. By identifying the most influential features, practitioners can better understand the patterns in the data that differentiate inliers from outliers.

Figure 4.12 illustrates the relative importance of the five main features of the Vowels dataset. The x-axis represents the importance scores, indicating the proportional influence of each feature on the predictions of the model, while the y-axis lists the feature identifiers.

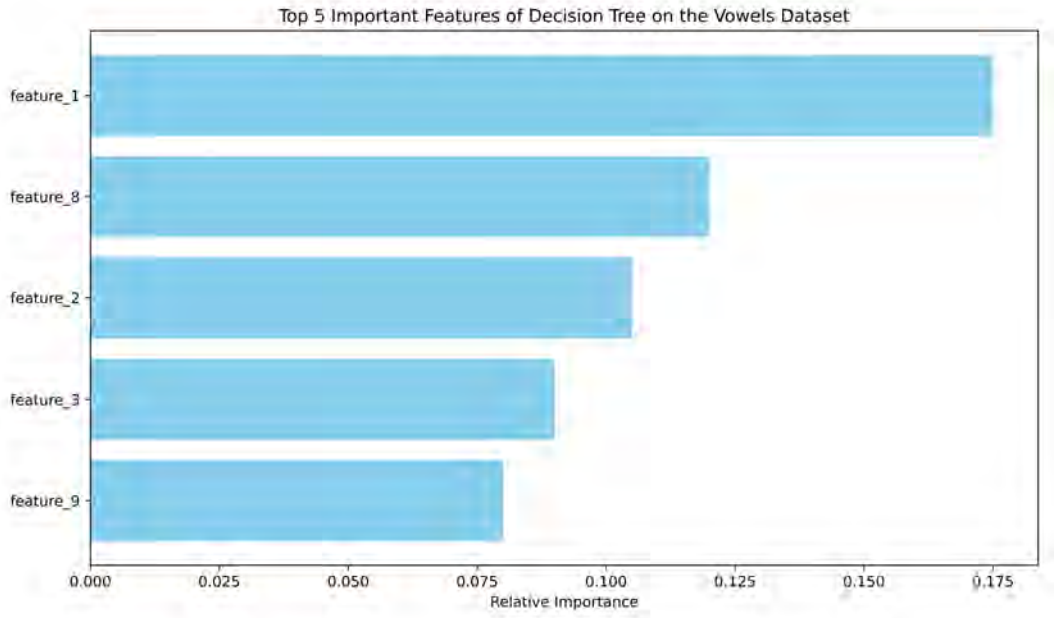

Figure 4.12: Relative importance of the top 5 features in the Vowels dataset as assessed by the decision tree model.

The analysis highlights feature_1 as the most critical factor, contributing approximately 17.5% to the model's decisions, suggesting its significant role in distinguishing outliers from inliers. Following this, feature_8 and feature_2 show notable contributions, with importance scores of 12.5% and 10%, respectively. These features play a substantial role in the decision-making process. Rounding out the top five are feature_3 and feature_9, with contributions of 9% and 7.5%, respectively, indicating their moderate, but meaningful influence on classification.

This analysis underscores the utility of feature importance scores in interpreting model outputs, enhancing transparency and trust in the model's decision-making process.

Practical Applications of Feature Importances

Analyzing the feature importance enables practitioners to identify the most impactful features, particularly in high-dimensional datasets, where many features may contribute little to the predictive power of the model. Features with negligible importance scores can be candidates for removal during feature selection, simplifying the model, reducing computational demands, and potentially improving its performance.

In domains such as healthcare or finance, where interpretability is essential, understanding the feature importance allows practitioners to explain the predictions of the model to stakeholders more effectively. If a decision tree highlights a feature as particularly influential, domain experts can investigate whether its importance aligns with established knowledge or if further analysis is needed to confirm its predictive validity and rule out potential artifacts.

For the Vowels dataset, highly important features like feature_1 and feature_8 warrant closer examination to understand their significant influence on classification outcomes. Investigating these features might reveal whether they represent unique acoustic characteristics or other distinctive attributes of vowel sounds. Such insights can guide better data collection strategies, improve feature engineering, or even contribute to domain-specific discoveries.

This approach underscores the dual value of feature importance analysis: Not

only does it improve model interpretability and trust, it also fosters opportunities to refine models and uncover meaningful patterns within the data.

Comparing Feature Importances with SHAP Explanations

To assess the quality of the feature importance rankings produced by our decision tree, we compared them with those derived from SHapley Additive exPlanations (SHAP) [288] applied to an XGBoost model [87]. XGBoost, a gradient-boosted decision tree ensemble, is renowned for its predictive performance but often sacrifices interpretability. SHAP provides a post hoc explanation framework that assigns feature importance scores by analyzing the marginal contributions of each feature across various combinations.

We trained an XGBoost model on the Vowels dataset and applied SHAP to identify the top 5 features (Figure 4.13). There was an 80% overlap between the top features identified by the decision tree and those ranked by SHAP, with features such as feature_1, feature_2, feature_3, and feature_8 appearing in both lists. This substantial overlap demonstrates the reliability of the decision tree's feature importance rankings, as they align closely with those derived from a more complex and widely used model.

The overlap highlights the decision tree's capability to capture the primary factors influencing predictions, even without the computational complexity of SHAP. While SHAP provides a global view of feature contributions by considering all possible feature interactions, the decision tree determines importance directly from its structure, based on metrics like split frequency and impurity reduction. Despite these methodological differences, the consistency between the two approaches validates the decision tree's ability to identify key features effectively and reinforces its utility as a transparent and interpretable model.

4.4.7.8 Local Interpretability through Decision Tree Visualization

Figure 4.14 visualizes the full decision tree used to classify data points as outliers or non-outliers in the lympho dataset, providing insights into both global and local interpretability. By following the path of a specific data point through the tree, from the root to its corresponding leaf node, we can understand the contributions of

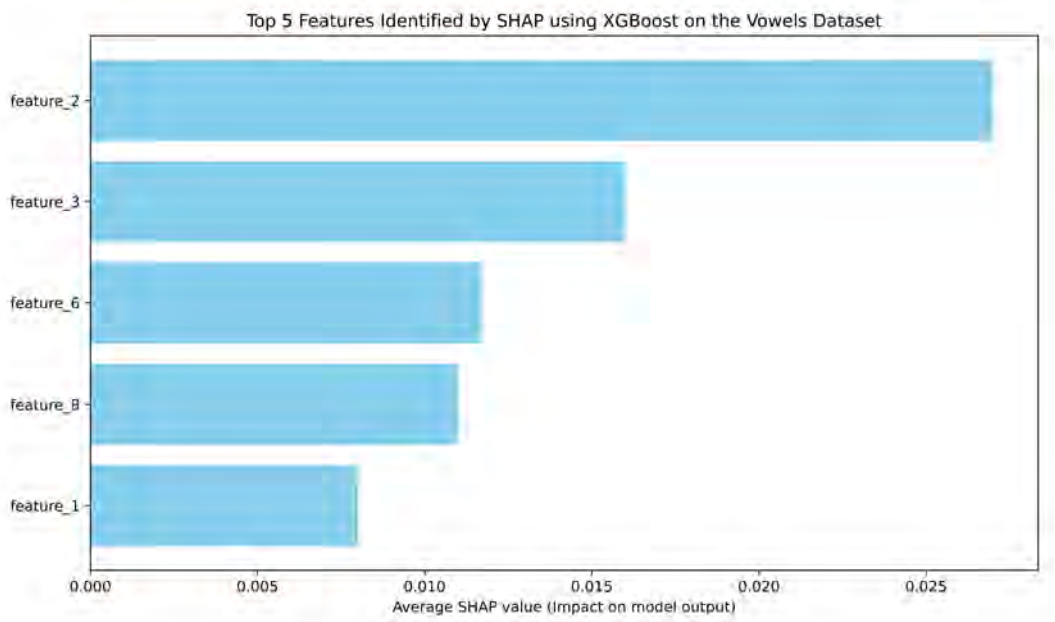

Figure 4.13: Top 5 features identified using SHapley Additive exPlanations (SHAP) on an XGBoost model trained on the Vowels dataset. SHAP values indicate the average contribution of each feature to the model's output, offering a post hoc explanation of feature importances in a complex model.

various features to the final classification decision.

At the root node, the tree is split according to the condition feature_18 ≤ 1.798, predominantly classifying samples as non-outliers if this condition is met. If the condition is not satisfied, the model evaluates subsequent features such as feature_14, feature_12, feature_1, feature_6, and feature_8, applying specific thresholds at each decision point. Each node in the tree displays the entropy value (indicating the purity of the split) and the count of samples in each category, offering a detailed view of the global logic of the model.

For local interpretability, the decision tree allows practitioners to track the exact path taken by any individual data point. This path outlines the sequence of feature-based decisions that led to its classification, providing a transparent explanation of the decision-making process. By examining these feature conditions and thresholds, users can identify the key factors influencing specific classifications, gaining valuable insights into both the overall behavior of the model and the reasons behind particular outcomes.

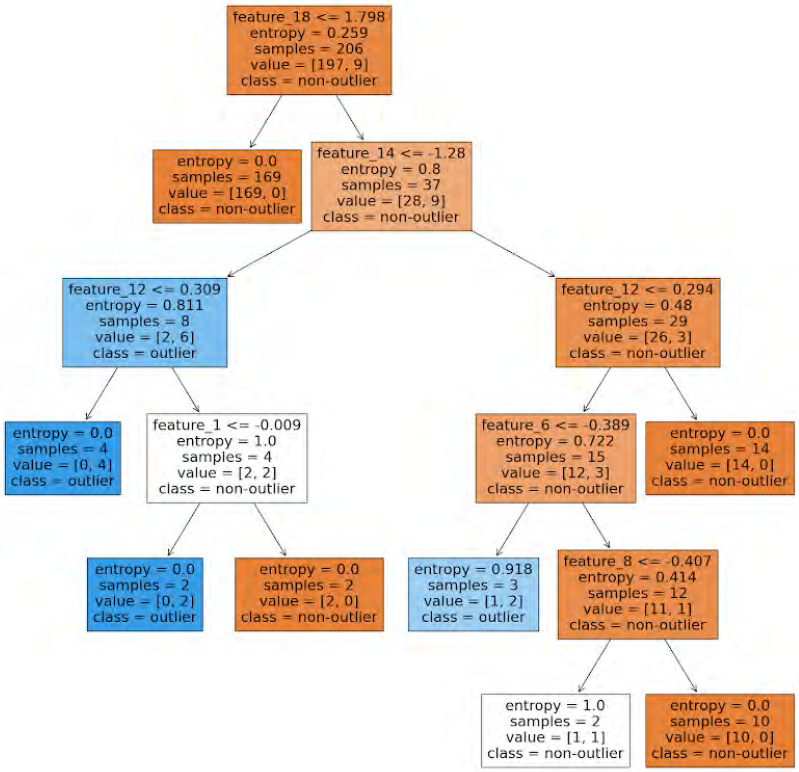

Figure 4.14: Decision tree used for outlier detection in the lympho dataset. Each node provides entropy values and sample counts, illustrating the model's global decision framework and its local interpretability for individual data points.

4.4.7.9 Decision Tree Rules

The rules extracted from the decision tree for the lympho dataset, illustrated in Figure 4.15, provide a streamlined and interpretable representation of the decision-making process of the model. These rules focus specifically on the conditions that lead to outlier classification, providing a clear and concise view of the criteria the model uses to identify outliers.

By isolating only the paths relevant to outlier classification, this representation minimizes clutter, making it easier for practitioners to debug and interpret the

Rule 1:	If feature_18 $>$ 1.799, proceed as follows:		
	Rule 1.1:	If feature_14 $\leq$ -1.28:	
		Rule 1.1.1:	If feature_12 $\leq$ 0.31, classify as outlier (**class: 1.0**).
		Rule 1.1.2:	If feature_12 $>$ 0.31 and feature_1 $\leq$ -0.01, classify as outlier (**class: 1.0**).
	Rule 1.2:	If feature_14 $>$ -1.28, feature_12 $\leq$ 0.29, and feature_6 $\leq$ -0.39, classify as outlier (**class: 1.0**).	
Note:	Data points that do not meet any of these conditions are classified as non-outliers (**class: 0.0**).		

Figure 4.15: Rules for identifying outliers in the lympho dataset, derived from the decision tree. Each rule specifies a sequence of feature-based conditions that lead to outlier classification.

model's decisions. This is particularly beneficial in environments where rapid and accurate decision-making is essential, as the rules highlight the most critical thresholds and feature interactions in the dataset.

4.5 Summary and Implications

This chapter presented a novel framework for interpretable unsupervised outlier detection, addressing the dual challenges of achieving high detection performance and ensuring model interpretability. By integrating normalizing flows with decision trees, the method combines the generative modeling capabilities of deep learning with the transparency of interpretable models, offering a robust solution for high-stakes domains such as fraud detection and healthcare, where accuracy and trust are critical.

Extensive evaluation across 17 datasets against 23 state-of-the-art methods demonstrated that the framework achieves competitive or superior performance across metrics such as precision, F1 score, and the Matthews correlation coefficient. In particular, the method excels at minimizing false positives, a crucial factor in sensitive applications. Decision trees provide both local and global interpretabil-

ity, allowing clear explanations for individual outlier detections as well as broader decision patterns. This transparency improves trust and facilitates practical adoption in real-world settings.

The framework bridges the gap between unsupervised and supervised learning by taking advantage of the explainability inherent in decision tree structures within an unsupervised context. This integration not only maintains strong performance, but also significantly enhances interpretability, enabling more responsible and trustworthy decision-making.

By addressing critical gaps in interpretability for unsupervised outlier detection, this work contributes to the broader objectives of explainable AI. The method offers a practical pathway for deploying interpretable unsupervised learning systems with confidence in high-stakes applications. These insights, together with the contributions of previous chapters, set the stage for the final chapter, which synthesizes the main findings of the thesis, acknowledges limitations, and explores directions for future research.

Chapter 5

Conclusion

This thesis has contributed to the field of unsupervised outlier detection by addressing challenges related to model selection and interpretability.

In Chapter 2, a meta-learning algorithm was introduced that leverages ensemble techniques and the selection of unsupervised spectral features to enhance outlier detection. This approach advances beyond simple aggregation of detector outputs by critically evaluating and selecting the most effective detectors for a given dataset. This selective process improves both the accuracy and efficiency of outlier detection systems, leading to a more robust model selection process, a crucial advancement in accuracy-critical environments. The effectiveness of this meta-learning algorithm was demonstrated on a variety of tabular datasets, showcasing its potential to improve the performance and reliability of unsupervised outlier detection in practical applications.

Chapter 3 presented a comprehensive taxonomy of machine learning interpretability methods. This taxonomy provides a structured framework for categorizing interpretability approaches based on key dimensions such as local versus global interpretability, model-agnostic versus model-specific techniques, and data type applicability. By highlighting the trade-offs among different methods, the taxonomy offers practical guidance for researchers and practitioners to select interpretability strategies tailored to their specific goals and constraints. This framework aims to facilitate the effective integration of interpretability into machine learning workflows, particularly in fields where transparency and accountability are paramount.

Chapter 4 introduced a novel approach to outlier detection that integrates normalizing flows with decision trees. This method effectively balances interpretability and detection accuracy by providing clear, human-understandable decision rules while maintaining strong outlier identification performance. This approach is particularly valuable in industries where transparent decision-making is crucial for building trust and ensuring reliability, as it allows for both accurate outlier detection and clear explanations of the underlying reasons for the classification. The effectiveness of this method was demonstrated on both tabular and image datasets, showcasing its potential to enhance interpretability with minimal impact on detection accuracy.

Future Directions

Although this thesis has addressed key challenges in unsupervised outlier detection, many open questions and opportunities for further investigation remain.

Building upon the meta-learning algorithm proposed in Chapter 2, future work could investigate its scalability to high-dimensional datasets and real-time applications to ensure its effectiveness in large-scale or time-sensitive settings. Moreover, adapting the framework to handle non-tabular data, such as time series, graph structures, or text, would necessitate incorporating mechanisms to account for the inherent temporal dependencies or structural relationships in such data. Finally, addressing concept drift, where data distributions change over time, could significantly improve the algorithm's adaptability and long-term performance in dynamic environments.

The taxonomy of interpretability methods developed in Chapter 3 could be further enhanced by addressing the specific interpretability challenges posed by unsupervised learning and multimodal data scenarios. Expanding the taxonomy to encompass these areas would better align it with the evolving needs of researchers and practitioners working with increasingly complex datasets and models. Incorporating guidance and best practices tailored to these challenges would further enhance the taxonomy's practical value and contribute to the development of more

interpretable and trustworthy machine learning systems across diverse domains.

The outlier detection method introduced in Chapter 4, which integrates normalizing flows with decision trees, could be further refined by addressing the observed trade-off between precision and recall. Improving the method's ability to detect all relevant outliers, particularly in datasets where recall is critical, could be achieved through adjustments to the normalizing flow architecture or by exploring alternative density estimation techniques. Another challenge lies in the computational complexity of training the normalizing flow model, which can be demanding for large-scale datasets. Simplifying the architecture or investigating alternative training methods could improve efficiency and scalability. Furthermore, extending the method's applicability to other data types, such as time series or graph data, and exploring its ability to address concept drift would broaden its utility in a wider range of real-world scenarios.

In summary, this thesis presented methods to address specific challenges in unsupervised outlier detection and interpretability. Future work should focus on scaling these methods to handle very high-dimensional and diverse data types, including time series, graph, and multimodal datasets, while improving computational efficiency to support large-scale and real-time applications. Addressing evolving data distributions through mechanisms such as concept drift adaptation and improving the balance between precision and recall could further refine these approaches. By integrating these advances, the methods developed in this thesis can be extended to broader contexts, fostering more robust, transparent, and adaptable machine learning systems.

Appendix A

Supplementary Data and Tables

Table A.1: Repository Links For Machine Learning Interpretability Tools.

Tool	Repository Link
accessorize-to-a-crime	https://github.com/mahmoods01/accessorize-to-a-crime
adversarial-squad	https://github.com/robinjia/adversarial-squad
adversarial_text	https://github.com/aonotas/adversarial_text
adversarial_training	https://github.com/WangJiuniu/adversarial_training
adversarial_training_methods	https://github.com/enry12/adversarial_training_methods
aequitas	https://github.com/dssg/aequitas
AIX360	https://github.com/Trusted-AI/AIX360
alibi	https://github.com/SeldonIO/alibi
AnalysisBySynthesis	https://github.com/bethgelab/AnalysisBySynthesis
Anchor	https://github.com/marcotcr/anchor
boundary-attack	https://github.com/greentfrapp/boundary-attack
CAM	https://github.com/zhoubolei/CAM
cleverhans	https://github.com/tensorflow/cleverhans
debiaswe	https://github.com/tolga-b/debiaswe
Deep Visualization Toolbox	https://github.com/yosinski/deep-visualization-toolbox
DeepExplain	https://github.com/marcoancona/DeepExplain

Continued on next page

Table A.1 – continued from previous page

Tool	Repository Link
DeepLift	https://github.com/kundajelab/deeplift
DLIME	https://github.com/rehmanzafar/dlime_experiments
Eli5	https://github.com/TeamHG-Memex/eli5
equalized_odds_and_calibration	https://github.com/gpleiss/equalized_odds_and_calibration
fair-classification	https://github.com/mbilalzafar/fair-classification
fairlearn	https://github.com/fairlearn/fairlearn
FairMachineLearning	https://github.com/jtcho/FairMachineLearning
fairml	https://github.com/adebayoj/fairml
fairness	https://github.com/dodger487/fairness
fairness-comparison	https://github.com/algofairness/fairness-comparison
fairness-in-ml	https://github.com/equialgo/fairness-in-ml
foolbox	https://github.com/bethgelab/foolbox
GerryFair	https://github.com/algowatchpenn/GerryFair
gnn-meta-attack	https://github.com/danielzuegner/gnn-meta-attack
Grad-CAM	https://github.com/ramprs/grad-cam
Grad-CAM ++	https://github.com/adityac94/Grad_CAM_plus_plus
graph_adversarial_attack	https://github.com/Hanjun-Dai/graph_adversarial_attack
HSJA	https://github.com/Jianbo-Lab/HSJA
influence-release	https://github.com/kohpangwei/influence-release
iNNvestigate	https://github.com/albermax/innvestigate
Integrated Gradients	https://github.com/ankurtaly/Integrated-Gradients
InterpretML	https://github.com/interpretml/interpret
L2X	https://github.com/Jianbo-Lab/L2X
lime	https://github.com/marcotcr/lime
ML-fairness-gym	https://github.com/google/ml-fairness-gym

Continued on next page

Table A.1 – continued from previous page

Tool	Repository Link
Nattack	https://github.com/gaussian-attack/Nattack
nettack	https://github.com/danielzuegner/nettack
nlp_adversarial_examples	https://github.com/nesl/nlp_adversarial_examples
nn_robust_attacks	https://github.com/carlini/nn_robust_attacks
one-pixel-attack-keras	https://github.com/Hyperparticle/one-pixel-attack-keras
PDPbox	https://github.com/SauceCat/PDPbox
procedurally_fair_learning	https://github.com/nina-gh/procedurally_fair_learning
pyBreakDown	https://github.com/MI2DataLab/pyBreakDown
PyCEbox	https://github.com/AustinRochford/PyCEbox
rationale	https://github.com/taolei87/rcnn/tree/master/code/rationale
RISE	https://github.com/eclique/RISE
scpn	https://github.com/miyyer/scpn
SALib	https://github.com/SALib/SALib
shap	https://github.com/slundberg/shap
Skater	https://github.com/oracle/Skater
Slim	https://github.com/ustunb/slim-python
stAdv	https://github.com/rakutentech/stAdv
tcav	https://github.com/tensorflow/tcav
TextAttack	https://github.com/QData/TextAttack
TextFooler	https://github.com/jind11/TextFooler
tf-explain	https://github.com/sicara/tf-explain
The LRP Toolbox	https://github.com/sebastian-lapuschkin/lrp_toolbox
themis-ml	https://github.com/cosmicBboy/themis-ml
transferability-advdnn-pub	https://github.com/sunblaze-ucb/transferability-advdnn-pub
UAN	https://github.com/jhayes14/UAN

Continued on next page

Table A.1 – continued from previous page

Tool	Repository Link
universal	https://github.com/LTS4/universal
WordAdver	https://github.com/AnyiRao/WordAdver
ZOO-Attack	https://github.com/huanzhang12/ZOO-Attack

Appendix B

Thesis Summary in Greek (Περίληψη της Διατριβής στα Ελληνικά)

Τίτλος

Ανίχνευση Ανωμαλιών Χωρίς Επίβλεψη: Νέες Προσεγγίσεις για Ισχυρές Ομάδες Ταξινομητών και Ερμηνευσιμότητα.

Σύνοψη

Η ανίχνευση ανωμαλιών κατέχει καίρια θέση σε πολλούς τομείς, όπως η αναγνώριση απάτης, η ασφάλεια δικτύων και η ιατρική διάγνωση. Ωστόσο, η διαδικασία επιλογής μοντέλου και η ερμηνευσιμότητα των αποτελεσμάτων αποτελούν σημαντικές προκλήσεις, ιδίως σε περιβάλλοντα μη επιβλεπόμενης μάθησης, όπου απουσιάζουν ετικετοποιημένα δεδομένα. Η παρούσα διατριβή επιδιώκει να δώσει απαντήσεις σε αυτές τις προκλήσεις, παρουσιάζοντας καινοτόμες μεθοδολογίες για την αποτελεσματικότερη επιλογή μοντέλου σε ομάδες ταξινομητών (ensembles) και για την ενίσχυση της ερμηνευσιμότητας.

Σε πρώτη φάση, εισάγεται μια μέθοδος μετα-μάθησης για την επιλογή μοντέλων σε ομάδες ταξινομητών ανωμαλιών. Η προσέγγιση αυτή αξιοποιεί τις προβλέψεις που παράγονται από πολλαπλούς επιμέρους ταξινομητές, θεωρώντας τες ως ένα νέο σύνολο χαρακτηριστικών. Στη συνέχεια, εφαρμόζει τεχνικές επιλογής χαρακτηριστικών χωρίς επίβλεψη σε αυτά τα μετα-χαρακτηριστικά, με στόχο να εντοπίσει τους ταξινομητές που αποδεικνύονται πιο συναφείς με την εγγενή δομή

του εκάστοτε συνόλου δεδομένων. Οι επιλεγμένοι ταξινομητές συνδυάζονται τελικά μέσω μιας στρατηγικής ψηφοφορίας, αξιοποιώντας έτσι τη συλλογική τους γνώση για την τελική ανίχνευση.

Σε δεύτερο επίπεδο, παρουσιάζεται μια λεπτομερής ταξινομία των διαθέσιμων μεθόδων ερμηνευσιμότητας. Η ταξινομία αυτή κατηγοριοποιεί τις υπάρχουσες τεχνικές με βάση κριτήρια όπως το εύρος της ερμηνείας (τοπική για μεμονωμένες προβλέψεις ή καθολική για ολόκληρο το μοντέλο), η εξάρτηση από το συγκεκριμένο μοντέλο (ανεξάρτητες ή ειδικές για έναν τύπο μοντέλου) και ο χρόνος εφαρμογής: είτε κατά την εκπαίδευση (εγγενείς μέθοδοι) είτε μετά την εκπαίδευση (μέθοδοι εκ των υστέρων). Επιπλέον, η ταξινομία λαμβάνει υπόψη την εφαρμοσιμότητα των μεθόδων σε διάφορους τύπους δεδομένων, όπως πίνακες, κείμενο, εικόνες και δεδομένα γραφημάτων.

Τέλος, προτείνεται μια νέα μέθοδος ανίχνευσης ανωμαλιών χωρίς επίβλεψη που συνδυάζει τα πλεονεκτήματα των ροών κανονικοποίησης στην εκτίμηση της πυκνότητας των δεδομένων και την ερμηνευσιμότητα των δέντρων αποφάσεων. Η μέθοδος χρησιμοποιεί ροές κανονικοποίησης για τη δημιουργία ενός συνθετικού συνόλου δεδομένων με ψευδο-ετικέτες, οι οποίες υποδεικνύουν πιθανές ανωμαλίες. Στη συνέχεια, αυτό το σύνολο δεδομένων αξιοποιείται για την εκπαίδευση ενός δέντρου απόφασης. Η προσέγγιση αυτή παρέχει σαφή όρια ταξινόμησης που βασίζονται σε κανόνες, διευκολύνοντας την κατανόηση των κριτηρίων με τα οποία εντοπίζονται οι ανωμαλίες.

Συνολικά, η παρούσα διατριβή συμβάλλει στην πρόοδο των μη επιβλεπόμενων συστημάτων ανίχνευσης ανωμαλιών μέσω της βελτίωσης της διαδικασίας επιλογής μοντέλου για πιο ισχυρές ομάδες ταξινομητών, της ενίσχυσης της ερμηνευσιμότητας των αποτελεσμάτων και της παροχής μιας πιο ξεκάθαρης κατανόησης της διαδικασίας λήψης αποφάσεων. Με αυτόν τον τρόπο, ενισχύεται η ευρύτερη και πιο αξιόπιστη χρήση αυτών των συστημάτων σε κρίσιμους τομείς της καθημερινότητας.

Εισαγωγή

Η ανίχνευση ανωμαλιών είναι η διαδικασία εντοπισμού παρατηρήσεων σε ένα σύνολο δεδομένων που ξεφεύγουν αισθητά από τα αναμενόμενα πρότυπα ή την κανονική συμπεριφορά. Αυτές οι ανώμαλες παρατηρήσεις, που συχνά οφείλονται σε διάφορες υποκείμενες αιτίες, συνήθως αποτελούν λιγότερο από το 5% των δεδομένων [1]. Αν και δεν υπάρχει ένας μοναδικός, παγκοσμίως αποδεκτός ορισμός για τις ανωμαλίες [2], η αναγνώρισή τους είναι θεμελιώδης για την αποτελεσματική ανάλυση δεδομένων. Μας επιτρέπει να εντοπίσουμε και, αν χρειαστεί, να αφαιρέσουμε αυτές τις παρατηρήσεις που θα μπορούσαν να επηρεάσουν αρνητικά ή να οδηγήσουν σε λανθασμένα συμπεράσματα.

Η ανίχνευση ανωμαλιών είναι υψίστης σημασίας σε πολλούς τομείς, όπου η εμφάνισή τους μπορεί να σηματοδοτεί σημαντικά ή επιζήμια γεγονότα. Για παράδειγμα, στον χρηματοπιστωτικό τομέα, βοηθά στην ανακάλυψη δόλιων συναλλαγών [3, 4], ενώ στην υγειονομική περίθαλψη, συμβάλλει στη διάγνωση ασθενειών, όπως ο καρκίνος, μέσω ιατρικών απεικονίσεων [5, 6]. Αντίστοιχα, στις τηλεπικοινωνίες χρησιμοποιείται για την αποκάλυψη απάτης, και στα βιομηχανικά συστήματα για την πρόβλεψη βλαβών εξοπλισμού, μειώνοντας έτσι τον δαπανηρό χρόνο διακοπής λειτουργίας [2]. Άλλες εφαρμογές περιλαμβάνουν τα συστήματα Internet of Things (IoT) [7], την παρακολούθηση της ακεραιότητας κατασκευών [8] και την ανίχνευση εισβολών σε δίκτυα, όπου συχνά αντιμετωπίζονται προκλήσεις όπως η επεξεργασία μεγάλου όγκου δεδομένων υψηλών διαστάσεων που μεταδίδονται συνεχώς [9, 10].

Από τη σκοπιά της επιστήμης των δεδομένων, η αποτελεσματική διαχείριση των ανωμαλιών διασφαλίζει την αξιοπιστία των μοντέλων μηχανικής μάθησης και των στατιστικών αναλύσεων, επιτρέποντας στους αλγορίθμους να αναγνωρίζουν με μεγαλύτερη ακρίβεια τις υποκείμενες τάσεις και να εξάγουν ουσιαστικές πληροφορίες [11, 12]. Παρόλο που οι ανωμαλίες μπορούν να παραπλανήσουν τα μοντέλα, ταυτόχρονα μπορούν να αποτελέσουν πολύτιμα σήματα, αποκαλύπτοντας απρόβλεπτες ή καινοτόμες συμπεριφορές ενός συστήματος [13]. Αυτός ο διπλός ρόλος περιπλέκει την ανίχνευσή τους: κάποιες ανωμαλίες πρέπει να απο-

μακρυνθούν για να διατηρηθεί η ποιότητα των δεδομένων, ενώ άλλες απαιτούν λεπτομερή ανάλυση για να κατανοήσουμε τις αιτίες τους. Σε τομείς όπου η ασφάλεια είναι κρίσιμη, όπως η ανίχνευση απάτης ή η παρακολούθηση της λειτουργίας συστημάτων, η διάκριση μεταξύ αυτών των δύο τύπων ανωμαλιών είναι απαραίτητη [14, 15].

Παρά τη μεγάλη της σημασία, η ανίχνευση ανωμαλιών εξακολουθεί να αποτελεί μια σημαντική ερευνητική πρόκληση, ιδιαίτερα όταν τα σύνολα δεδομένων είναι υψηλών διαστάσεων, περιέχουν θόρυβο και αλλάζουν με την πάροδο του χρόνου. Η ανάπτυξη ισχυρών μεθοδολογιών είναι απαραίτητη για την ακριβή αναγνώριση και κατανόηση των ανωμαλιών, ώστε να μπορούν να εξαχθούν χρήσιμες πληροφορίες και να βελτιωθεί η απόδοση των μοντέλων.

Τα πλαίσια μηχανικής μάθησης που χρησιμοποιούνται για την ανίχνευση ανωμαλιών μπορούν να κατηγοριοποιηθούν σε επιβλεπόμενες, ημι-επιβλεπόμενες και μη-επιβλεπόμενες μεθόδους, καθεμία από τις οποίες έχει τα δικά της πλεονεκτήματα και μειονεκτήματα [2, 16]. Η επιλογή της κατάλληλης μεθόδου εξαρτάται συχνά από τη διαθεσιμότητα ετικετοποιημένων δεδομένων και τα χαρακτηριστικά του συγκεκριμένου συνόλου δεδομένων.

Η ανίχνευση ανωμαλιών είναι ένα κρίσιμο ζήτημα στην ανάλυση δεδομένων, ειδικά όταν δεν υπάρχουν διαθέσιμα ετικετοποιημένα δεδομένα. Παρά την ανάπτυξη πολλών μη επιβλεπόμενων μεθόδων, υπάρχουν αρκετές προκλήσεις που δυσχεραίνουν την ευρεία πρακτική τους εφαρμογή και αποτελούν κίνητρο για περαιτέρω έρευνα. Η παρούσα διατριβή παρουσιάζει τρεις κύριες συνεισφορές στον τομέα της μη επιβλεπόμενης ανίχνευσης ανωμαλιών, καθεμία από τις οποίες στοχεύει στην αντιμετώπιση των περιορισμών που αναφέρθηκαν παραπάνω και έχει δημοσιευτεί σε επιστημονικά περιοδικά.

1. Ένα πλαίσιο μετα-μάθησης για την επιλογή μοντέλων σε ομάδες ταξινομητών: Αυτό το πλαίσιο αξιοποιεί την επιλογή χαρακτηριστικών για να αναγνωρίσει τα μοντέλα ανίχνευσης ανωμαλιών που παρέχουν την περισσότερη πληροφορία, βελτιώνοντας την ανθεκτικότητα και την ακρίβεια μέσω του συνδυασμού αυτών των επιλεγμένων μοντέλων.

2. Μια ταξινομία μεθόδων ερμηνευσιμότητας μηχανικής μάθησης: Αυτή η ταξινομία κατηγοριοποιεί τις τεχνικές ερμηνευσιμότητας με βάση το εύρος, την εξάρτηση από το μοντέλο, τον χρόνο εφαρμογής και τον τύπο δεδομένων, βοηθώντας στην επιλογή κατάλληλων μεθόδων.

3. Μια ερμηνεύσιμη μέθοδος ανίχνευσης ανωμαλιών: Αυτή η μέθοδος συνδυάζει ροές κανονικοποίησης για εκτίμηση πυκνότητας με δέντρα αποφάσεων για ερμηνευσιμότητα που βασίζεται σε κανόνες, παρέχοντας τόσο ακριβή ανίχνευση όσο και σαφείς διαδικασίες λήψης αποφάσεων.

Ισχυρές Ομάδες Ταξινομητών: Μια Μέθοδος Μετα-Μάθησης για Επιλογή Μοντέλων Χωρίς Επίβλεψη

Αυτό το κεφάλαιο πραγματεύεται σε βάθος το πρόβλημα της επιλογής μοντέλου στην ανίχνευση ανωμαλιών χωρίς επίβλεψη. Η επιλογή μοντέλου είναι μια κρίσιμη πτυχή στην ανάπτυξη ισχυρών συστημάτων ανίχνευσης ανωμαλιών, καθώς η απόδοση μπορεί να διαφέρει σημαντικά ανάλογα με τον επιλεγμένο αλγόριθμο και τις ιδιαιτερότητες του εκάστοτε συνόλου δεδομένων. Για να αντιμετωπιστεί αυτή η πρόκληση, παρουσιάζεται ένα προηγμένο πλαίσιο μετα-μάθησης, ειδικά σχεδιασμένο για να ενισχύσει την ανθεκτικότητα και την ακρίβεια των ομάδων ταξινομητών που χρησιμοποιούνται για την ανίχνευση ανωμαλιών.

Το προτεινόμενο πλαίσιο χρησιμοποιεί την επιλογή χαρακτηριστικών χωρίς επίβλεψη (unsupervised feature selection) ως έναν έξυπνο μηχανισμό για την αναγνώριση και ιεράρχηση των μεμονωμένων μοντέλων ανίχνευσης ανωμαλιών. Με αυτόν τον τρόπο, εντοπίζονται τα μοντέλα που παρέχουν τις περισσότερες πληροφορίες σχετικά με την υποκείμενη δομή ενός συνόλου δεδομένων. Η ικανότητα διάκρισης μεταξύ των αλγορίθμων που είναι πιο συμβατοί με τη δομή του εκάστοτε συνόλου δεδομένων και αυτών που είναι λιγότερο αξιόπιστοι είναι ζωτικής σημασίας για τη δημιουργία ενός ισχυρού συνόλου ταξινομητών. Επιλέγοντας στρατηγικά τα πιο κατάλληλα μοντέλα για το εκάστοτε σύνολο δεδομένων, το πλαίσιο μειώνει αποτελεσματικά την επίδραση των μοντέλων που μπορεί να είναι ευαίσθητα στον θόρυβο ή ακατάλληλα για συγκεκριμένα χαρακτηριστικά του

συνόλου δεδομένων. Αυτή η διαδικασία οδηγεί στη δημιουργία ενός πιο προσαρμόσιμου και αποτελεσματικού συστήματος για την ανίχνευση ανωμαλιών σε ένα ευρύ φάσμα συνόλων δεδομένων, το καθένα με τις δικές του μοναδικές στατιστικές ιδιότητες και προκλήσεις.

Η βασική μεθοδολογία του πλαισίου ξεκινά με την εκπαίδευση ενός συνόλου πολλαπλών βασικών μοντέλων ανίχνευσης ανωμαλιών. Κάθε ένα από αυτά αναλύει τα δεδομένα ανεξάρτητα και παράγει βαθμολογίες εμπιστοσύνης ή σκορ ανωμαλίας. Αυτά τα σκορ χρησιμεύουν ως ποσοτικοποίηση του βαθμού στον οποίο κάθε παρατήρηση θεωρείται «ανώμαλη» από το συγκεκριμένο μοντέλο. Στη συνέχεια, αυτές οι βαθμολογίες αντιμετωπίζονται ως χαρακτηριστικά για μια επακόλουθη διαδικασία επιλογής χαρακτηριστικών.

Συγκεκριμένα, για την επιλογή χαρακτηριστικών, το πλαίσιο αξιοποιεί τη Φασματική Επιλογή Χαρακτηριστικών (SPEC). Η μέθοδος SPEC είναι μια τεχνική επιλογής χαρακτηριστικών που στηρίζεται στις αρχές της φασματικής θεωρίας γραφημάτων. Η βασική ιδέα πίσω από το SPEC είναι να αξιολογήσει πόσο ταιριάζει το διάνυσμα βαθμολογίας που παράγει κάθε μοντέλο με την εγγενή δομή του εκάστοτε συνόλου δεδομένων [66]. Αυτό επιτυγχάνεται κατασκευάζοντας ένα γράφημα, όπου οι κόμβοι αντιπροσωπεύουν τις παρατηρήσεις και οι ακμές εκφράζουν την ομοιότητά τους. Στη συνέχεια, αναλύονται οι ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα Laplace αυτού του γραφήματος, ώστε να προσδιοριστούν τα χαρακτηριστικά (στην περίπτωσή μας, οι βαθμολογίες των μοντέλων) που είναι πιο συμβατά με τη δομή των δεδομένων. Αυτή η διαδικασία οδηγεί στην αναγνώριση και διατήρηση των μοντέλων που αντανακλούν καλύτερα τις σχέσεις μεταξύ των παρατηρήσεων, μειώνοντας ταυτόχρονα την επιρροή των λιγότερο σχετικών ή θορυβωδών μοντέλων. Έτσι, βελτιώνεται η ανθεκτικότητα της ομάδας ταξινομητών που προκύπτει, ιδίως σε σύνολα δεδομένων υψηλών διαστάσεων.

Η μαθηματική θεμελίωση της φασματικής επιλογής χαρακτηριστικών στο πλαίσιο αυτό περιλαμβάνει αρχικά την κατασκευή ενός γραφήματος ομοιότητας $W \in \mathbb{R}^{n \times n}$, όπου n είναι ο αριθμός των παρατηρήσεων. Η ομοιότητα μεταξύ δύο παρατηρήσεων s_i και s_j ποσοτικοποιείται χρησιμοποιώντας τη συνάρτηση πυρήνα

Radial Basis Function (RBF):

$$W_{ij} = \exp\left(-\frac{|s_i - s_j|^2}{2\sigma^2}\right). \tag{B.1}$$

όπου $|s_i - s_j|^2$ αντιπροσωπεύει την τετραγωνική ευκλείδεια απόσταση μεταξύ των παρατηρήσεων και σ είναι μια παράμετρος κλίμακας που ελέγχει την εμβέλεια της συνάρτησης πυρήνα. Στη συνέχεια, υπολογίζεται ο μη κανονικοποιημένος πίνακας Laplace $L = D - W$, όπου D είναι ο διαγώνιος πίνακας βαθμών του γραφήματος, με $D_{ii} = \sum_j W_{ij}$. Ακολουθεί η επίλυση του γενικευμένου προβλήματος ιδιοτιμών για τον κανονικοποιημένο πίνακα Laplace $L_{\text{norm}} = D^{-1/2} L D^{-1/2}$:

$$L_{\text{norm}} v = \lambda v, \tag{B.2}$$

όπου λ είναι οι ιδιοτιμές και v τα αντίστοιχα ιδιοδιανύσματα. Στο πλαίσιο της επιλογής χαρακτηριστικών με το SPEC για $k = 2$ συμπλέγματα, το δεύτερο μικρότερο ιδιοδιάνυσμα (το διάνυσμα Fiedler v_2) παίζει καθοριστικό ρόλο. Τα στοιχεία αυτού του διανύσματος χρησιμοποιούνται για την κατάταξη των αρχικών χαρακτηριστικών (δηλαδή των βαθμολογιών των μοντέλων), και επιλέγονται τα χαρακτηριστικά που αντιστοιχούν στις μεγαλύτερες απόλυτες τιμές στο διάνυσμα Fiedler, καθώς αυτά θεωρούνται ότι είναι τα πιο συνεπή με τον διαχωρισμό των δεδομένων σε δύο συμπλέγματα (κανονικές παρατηρήσεις και ανωμαλίες).

Στη συνέχεια, οι βαθμολογίες των μοντέλων που επιλέχθηκαν μέσω της διαδικασίας SPEC συνδυάζονται για να παραχθεί μια τελική ενιαία βαθμολογία ανωμαλίας για κάθε παρατήρηση. Για τον συνδυασμό αυτών των βαθμολογιών, το πλαίσιο χρησιμοποιεί τη στρατηγική Average of Maximum (AOM) [13]. Η βασική ιδέα πίσω από τη στρατηγική AOM είναι να χωρίσει το σύνολο των επιλεγμένων μοντέλων σε g υποομάδες. Για κάθε παρατήρηση, υπολογίζεται η μέγιστη βαθμολογία ανωμαλίας που έχει αποδοθεί από οποιοδήποτε μοντέλο εντός κάθε υποομάδας. Τέλος, η τελική βαθμολογία ανωμαλίας για την εν λόγω παρατήρηση προκύπτει από τον μέσο όρο αυτών των g μέγιστων βαθμολογιών. Μαθηματικά,

η βαθμολογία ανωμαλίας $S_{\mathrm{AOM}}(x_i)$ για μια παρατήρηση x_i υπολογίζεται ως:

$$S_{\mathrm{AOM}}(x_i) = \frac{1}{g}\sum_{j=1}^{g}\max_{d\in G_j} S_d(x_i). \tag{B.3}$$

όπου G_j αναπαριστά το σύνολο των μοντέλων στην j-οστή υποομάδα και $S_d(x_i)$ είναι η βαθμολογία ανωμαλίας που αποδίδεται στην παρατήρηση x_i από το μοντέλο d.

Η συγκεκριμένη αυτή προσέγγιση συνδυασμού επιδιώκει να εξισορροπήσει την επιρροή των ακραίων σημάτων ανωμαλίας που μπορεί να προέρχονται από μεμονωμένα μοντέλα με την ενσωμάτωση της ποικιλομορφίας των βαθμολογιών που παρέχονται από διαφορετικά μοντέλα. Αυτή η στρατηγική εξασφαλίζει μεγαλύτερη ανθεκτικότητα στον θόρυβο και στις πιθανές διακυμάνσεις στην κατανομή των δεδομένων, καθώς η τελική πρόβλεψη δεν βασίζεται υπερβολικά στην απόδοση ενός μόνο μοντέλου. Επιπλέον, η χρήση της παραλλαγής Fixed SPEC, η οποία κάνει την υπόθεση ότι τα δεδομένα μπορούν να διαχωριστούν σε δύο κύρια συμπλέγματα ($k = 2$), που αντιστοιχούν σε κανονικές παρατηρήσεις και ανωμαλίες, συμβάλλει στην αποτελεσματική μείωση του θορύβου κατά τη διαδικασία επιλογής μοντέλων, εστιάζοντας την προσοχή στα μοντέλα που παρέχουν τις πιο αξιόπιστες ενδείξεις για αυτόν τον διαχωρισμό. Η στρατηγική συνδυασμού βαθμολογιών AOM, μέσω του υπολογισμού της μέγιστης βαθμολογίας σε κάθε υποομάδα και στη συνέχεια του μέσου όρου αυτών των μέγιστων τιμών, επιτυγχάνει μια ισορροπία μεταξύ της ευαισθησίας στα ισχυρά σήματα ανωμαλίας και της συνολικής σταθερότητας της ομάδας ταξινομητών, μειώνοντας τον κίνδυνο ψευδώς θετικών αποτελεσμάτων και αξιοποιώντας αποτελεσματικά την ποικιλομορφία των υποκείμενων μοντέλων, ενισχύοντας έτσι την αξιοπιστία του συνόλου σε διάφορα και πιθανώς μεταβαλλόμενα περιβάλλοντα δεδομένων.

Συνοπτικά, αυτό το καινοτόμο πλαίσιο μετα-μάθησης αντιμετωπίζει αποτελεσματικά τις σημαντικές προκλήσεις που σχετίζονται με την ανίχνευση ανωμαλιών χωρίς επίβλεψη μέσω μιας διαδικασίας επιλογής μοντέλων που βασίζεται

στη φασματική ανάλυση και ενός «έξυπνου» συνδυασμού των επιλεγμένων μοντέλων. Οι πειραματικές αξιολογήσεις έδειξαν ότι το προτεινόμενο πλαίσιο επιδεικνύει βελτιωμένη απόδοση στην ανίχνευση ανωμαλιών σε σύγκριση με τυπικές και ευρέως χρησιμοποιούμενες τεχνικές. Ωστόσο, είναι σημαντικό να αναγνωριστεί ότι οι πραγματικές εφαρμογές συχνά θέτουν αυξημένες απαιτήσεις για μεγαλύτερη διαφάνεια και ερμηνευσιμότητα των συστημάτων ανίχνευσης ανωμαλιών. Η εγγενής πολυπλοκότητα του προτεινόμενου πλαισίου μετα-μάθησης, αν και ευεργετική για την απόδοση, μπορεί ενδεχομένως να συσκοτίσει τη λεπτομερή διαδικασία λήψης αποφάσεων, καθιστώντας δύσκολη την πλήρη κατανόηση των λόγων πίσω από μια συγκεκριμένη πρόβλεψη ανωμαλίας. Αυτή η έλλειψη διαφάνειας μπορεί να είναι ιδιαίτερα προβληματική σε κρίσιμες εφαρμογές όπου η οικοδόμηση εμπιστοσύνης και η συμμόρφωση με ρυθμιστικά και ηθικά πρότυπα είναι υψίστης σημασίας.

Συνεπώς, το κεφάλαιο ολοκληρώνεται με μια φυσική μετάβαση στο κρίσιμο θέμα της ερμηνευσιμότητας (interpretability) στην μηχανική μάθηση. Στο επόμενο μέρος, θα παρουσιαστεί μια αναλυτική ταξινομία διαφόρων μεθόδων ερμηνευσιμότητας, οι οποίες μπορούν να εφαρμοστούν για τη βελτίωση της διαφάνειας στα συστήματα μηχανικής μάθησης, συμπεριλαμβανομένων και αυτών που χρησιμοποιούνται για την ανίχνευση ανωμαλιών. Η επιδίωξη υψηλής απόδοσης στα μοντέλα μηχανικής μάθησης συχνά έρχεται σε αντίθεση με την εξίσου σημαντική ανάγκη για επεξηγησιμότητα και κατανόηση των αποφάσεών τους. Ο απώτερος στόχος της έρευνας στην ερμηνευσιμότητα είναι να γεφυρώσει αυτό το χάσμα, διασφαλίζοντας έτσι τις ηθικές απαιτήσεις που είναι απαραίτητες για την ευρεία και υπεύθυνη υιοθέτησή τους στον πραγματικό κόσμο.

Μία ταξινομία μεθόδων ερμηνευσιμότητας μηχανικής μάθησης

Καθώς τα μοντέλα γίνονται πιο σύνθετα, η κατανόηση της διαδικασίας λήψης αποφάσεών τους αποκτά κρίσιμη σημασία. Ενώ σύνθετα μοντέλα, όπως τα βαθιά νευρωνικά δίκτυα (deep neural networks) και οι ομάδες ταξινομητών, επιτυγχά-

νουν υψηλή απόδοση, συχνά λειτουργούν ως «μαύρα κουτιά» (black box) [85–87]. Αντίθετα, απλούστερα μοντέλα «λευκού κουτιού» (white box), όπως η γραμμική παλινδρόμηση [88] και τα δέντρα αποφάσεων [89], προσφέρουν διαφάνεια αλλά ενδέχεται να υστερούν σε προγνωστική ικανότητα. Η εξισορρόπηση της απόδοσης με την ερμηνευσιμότητα αποτελεί πλέον βασική απαίτηση για τα συστήματα AI, ιδίως σε κρίσιμες εφαρμογές. Η Επεξηγήσιμη Τεχνητή Νοημοσύνη (Explainable Artificial Intelligence - XAI) αντιμετωπίζει αυτό το ζήτημα ενισχύοντας τη διαφάνεια των μοντέλων, γεφυρώνοντας το χάσμα μεταξύ απόδοσης και ερμηνευσιμότητας [90].

Ανταποκρινόμενο σε αυτή την ανάγκη, το παρόν κεφάλαιο παρουσιάζει μια ταξινομία των μεθόδων ερμηνευσιμότητας μηχανικής μάθησης, προσφέροντας μια δομημένη επισκόπηση αυτού του ταχέως αναπτυσσόμενου τομέα. Οι κύριες συνεισφορές περιλαμβάνουν: (1) μια νέα ταξινομία των μεθόδων ερμηνευσιμότητας, κατηγοριοποιημένη ανά λειτουργική προσέγγιση και τομέα εφαρμογής, (2) λεπτομερή ανάλυση των υπαρχουσών τεχνικών, που περιγράφει τα πλεονεκτήματα και τα μειονεκτήματά τους, (3) συγκριτική ανάλυση που παρέχει πρακτικές οδηγίες για τη διαδικασία επιλογής μοντέλου με παραδείγματα, και (4) υπερσυνδέσμους σε προγραμματιστικές υλοποιήσεις και επιπλέον πόρους για την υποστήριξη της ενσωμάτωσής τους σε ροές εργασίας. Συνδέοντας τη θεωρητική πρόοδο με τις πρακτικές χρήσεις, το κεφάλαιο αυτό στοχεύει να επιτρέψει σε ερευνητές και επαγγελματίες να αναπτύξουν ερμηνεύσιμα μοντέλα μηχανικής μάθησης που είναι αξιόπιστα, διαφανή και αποτελεσματικά.

Αρχικά, το κεφάλαιο διευκρινίζει τους όρους «ερμηνευσιμότητα» και «επεξηγησιμότητα» (explainability). Παρότι συχνά χρησιμοποιούνται εναλλακτικά, οι ερευνητές έχουν προσπαθήσει να τους διακρίνουν [91–93]. Και οι δύο στερούνται ακριβών μαθηματικών ορισμών [94]. Η ερμηνευσιμότητα ορίζεται ως η ικανότητα εξήγησης ή παρουσίασης ενός μοντέλου σε όρους κατανοητούς για έναν άνθρωπο [92] και ο βαθμός στον οποίο ένα άτομο μπορεί να κατανοήσει την αιτία μιας απόφασης [95]. Με άλλα λόγια, η ερμηνευσιμότητα εστιάζει στη διαισθητική κατανόηση των σχέσεων εισόδου-εξόδου από τους χρήστες [94], ενώ η επεξ-

ηγησιμότητα εμβαθύνει στην εσωτερική λειτουργία ενός μοντέλου. Ένα μοντέλο μπορεί να είναι ερμηνεύσιμο χωρίς να είναι πλήρως επεξηγήσιμο, αλλά η επεξηγησιμότητα (explainability) συχνά απαιτεί την κατανόηση των εσωτερικών διεργασιών του μοντέλου [93]. Στην παρούσα εργασία, υιοθετείται η άποψη ότι η ερμηνευσιμότητα είναι μια ευρύτερη έννοια που περιλαμβάνει την επεξηγησιμότητα [92].

Στη συνέχεια, το κεφάλαιο εξετάζει την αξιολόγηση της ερμηνευσιμότητας μηχανικής μάθησης, περιγράφοντας τρεις κύριες προσεγγίσεις: την αξιολόγηση βάσει εφαρμογής (application-based evaluation), την αξιολόγηση βάσει ανθρώπου (human-based evaluation) και τη λειτουργικά θεμελιωμένη αξιολόγηση (functionally grounded evaluation) [92]. Η αξιολόγηση βάσει εφαρμογής εκτιμά την ικανότητα μιας μεθόδου ερμηνευσιμότητας να υποστηρίζει συγκεκριμένες εργασίες για τους τελικούς χρήστες. Η αξιολόγηση βάσει ανθρώπου περιλαμβάνει τη συμμετοχή κοινών χρηστών στην αξιολόγηση της κατανοησιμότητας των εξηγήσεων. Η λειτουργικά θεμελιωμένη αξιολόγηση χρησιμοποιεί μαθηματικούς ορισμούς χωρίς ανθρώπινη δοκιμή, καθιστώντας τη χρήσιμη όταν η ανθρώπινη αξιολόγηση είναι πρακτικά αδύνατη. Κάθε προσέγγιση εξετάζει διαφορετικές πτυχές της ερμηνευσιμότητας, απαιτώντας προσαρμοσμένες στρατηγικές αξιολόγησης.

Το κεφάλαιο εξετάζει υπάρχουσες εργασίες που ταξινομούν τις τεχνικές ερμηνευσιμότητας. Οι Gilpin et al. [93] πρότειναν μια ταξινομία για την ερμηνευσιμότητα νευρωνικών δικτύων. Οι Adadi και Berrada [94] διεξήγαγαν μια εκτενή ανασκόπηση, υποστηρίζοντας την ανάγκη για μεγαλύτερη τυποποίηση στην XAI. Οι Guidotti et al. [96] ταξινόμησαν τις τεχνικές ανάλογα με τα προβλήματα που αντιμετωπίζουν, επισημαίνοντας την έλλειψη τυπικών μετρικών αξιολόγησης. Οι Murdoch et al. [97] εισήγαγαν το πλαίσιο Προγνωστικό, Περιγραφικό, Σχετικό (Predictive, Descriptive, Relevant - PDR) για την αξιολόγηση. Οι Arrieta et al. [98] διέκριναν μεταξύ διαφανών και post-hoc μεθόδων, προτείνοντας μια εξειδικευμένη ταξινομία για την ερμηνευσιμότητα βαθιάς μάθησης.

Έπειτα, το κεφάλαιο εισάγει την προτεινόμενη ταξινομία του, τονίζοντας ότι οι μέθοδοι ερμηνευσιμότητας μπορούν να κατηγοριοποιηθούν ανάλογα με

1. τον σκοπό τους: (1) εξήγηση σύνθετων μοντέλων «μαύρου κουτιού», (2) δημιουργία διαφανών μοντέλων «λευκού κουτιού», (3) προώθηση της δικαιοσύνης (fairness) και μείωση των διακρίσεων, και (4) ανάλυση της ευαισθησίας

2. τη συμβατότητά τους με τους αλγορίθμους: ειδικές για ένα μοντέλο (model-specific) έναντι ανεξάρτητων από το μοντέλο (model-agnostic)

3. το εύρος της εξήγησης: τοπική (local) έναντι καθολικής (global)

4. τον τύπο των δεδομένων που χειρίζονται: πινακοειδή (tabular), εικόνα (image), κείμενο (text), γράφημα (graph) (sensitivity) των προβλέψεων του μοντέλου.

Αυτή η κατηγοριοποίηση στοχεύει στην παροχή ενός πρακτικού πλαισίου για την επιλογή τεχνικών ερμηνευσιμότητας που είναι κατάλληλες για συγκεκριμένους στόχους και δεδομένα.

Για την εξήγηση των μοντέλων «μαύρου κουτιού», συζητά τεχνικές οπτικοποίησης όπως το Grad-CAM [109] και τα αποσυνελικτικά δίκτυα (deconvolutional networks) [105] για την κατανόηση των ενεργοποιήσεων των νευρωνικών δικτύων. Επιπλέον, καλύπτει μεθόδους ανεξάρτητες από το μοντέλο, όπως το LIME [119] και το SHAP [123], καθώς και απλούστερα εργαλεία όπως τα διαγράμματα μερικής εξάρτησης (partial dependency plots - PDP) [121].

Όσον αφορά τα διαφανή μοντέλα «λευκού κουτιού», το κεφάλαιο επισημαίνει την εγγενή τους ερμηνευσιμότητα, αλλά συχνά τη χαμηλότερη προγνωστική ακρίβεια σε σύνθετες εργασίες. Τεχνικές όπως τα γενικευμένα προσθετικά μοντέλα (generalized additive models - GAMs) με αλληλεπιδράσεις ζευγών (pairwise interactions) (GA2Ms) [136], βασισμένα στην εργασία των Lou et al. [232], αναφέρονται, αλλά έχουν περιορισμένη εφαρμογή σε διαφορετικούς τύπους δεδομένων.

Στο πλαίσιο της δικαιοσύνης και της καταπολέμησης της προκατάληψης, το κεφάλαιο τονίζει την αυξανόμενη σημασία του σε ευαίσθητες εφαρμογές και αναφέρει πλαίσια όπως αυτό των Hardt et al. [162]. Ωστόσο, επισημαίνει ότι

μεγάλο μέρος αυτής της έρευνας επικεντρώνεται σε πινακοειδή δεδομένα, με περιορισμένη πρόοδο σε άλλους τύπους δεδομένων.

Για την ανάλυση ευαισθησίας και την ανθεκτικότητα σε αντιπαραθετικά παραδείγματα (adversarial examples), το κεφάλαιο υπογραμμίζει την ταχεία ανάπτυξη σε αυτόν τον τομέα μετά την ανακάλυψη των αντιπαραθετικών παραδειγμάτων [184], με αυξανόμενη έρευνα για την άμυνα και την ανθεκτικότητα.

Το κεφάλαιο αναλύει επίσης τις προκλήσεις στην υιοθέτηση της XAI, όπως η ασυνεπής ορολογία, η έλλειψη τυποποίησης, η περιορισμένη χρησιμότητα στον πραγματικό κόσμο και το χάσμα μεταξύ της έρευνας και των πρακτικών αναγκών. Πολλές υπάρχουσες τεχνικές δεν ανταποκρίνονται στις ειδικές απαιτήσεις της βιομηχανίας ή δεν διαχειρίζονται σύνθετα δεδομένα. Η αντιμετώπιση αυτών των προκλήσεων είναι ζωτικής σημασίας για την ευρύτερη εφαρμογή της XAI.

Συνοψίζοντας, το παρόν κεφάλαιο παρουσιάζει μια ολοκληρωμένη ταξινομία των μεθόδων ερμηνευσιμότητας στη μηχανική μάθηση, προσφέροντας έναν δομημένο τρόπο κατανόησης και σύγκρισης διαφορετικών προσεγγίσεων. Αυτή η ταξινομία αναδεικνύει ένα σημαντικό κενό στην ερμηνευσιμότητα των μοντέλων μάθησης χωρίς επίβλεψη, και ειδικότερα στην ανίχνευση ανωμαλιών χωρίς επίβλεψη. Παρότι η ερμηνευσιμότητα για μεθόδους μάθησης με επίβλεψη έχει σημειώσει πρόοδο, οι μη επιβλεπόμενες μέθοδοι συχνά στερούνται διαφάνειας, εμποδίζοντας την εμπιστοσύνη και την λογοδοσία. Για την αντιμετώπιση αυτού του ζητήματος, το επόμενο κεφάλαιο θα παρουσιάσει μια νέα, ερμηνεύσιμη μέθοδο για την ανίχνευση ανωμαλιών χωρίς επίβλεψη, με στόχο την αύξηση της διαφάνειας και της εμπιστοσύνης σε αυτόν τον τομέα.

Ερμηνεύσιμη Ανίχνευση Ανωμαλιών: Μια Μέθοδος Συνδυασμού Ροών Κανονικοποίησης και Δέντρων Αποφάσεων

Η ερμηνευσιμότητα στη μάθηση χωρίς επίβλεψη, ιδίως στην ανίχνευση ανωμαλιών, υστερεί συχνά σε σύγκριση με την επιβλεπόμενη [28]. Η έλλειψη αυτή δυσχεραίνει την ανάπτυξη και την υιοθέτηση τέτοιων μεθόδων, ιδίως σε κρίσιμους τομείς όπως η υγεία και η ασφάλεια, όπου η εμπιστοσύνη και η επεξήγηση

των αποτελεσμάτων είναι ύψιστης σημασίας [29].

Για την αντιμετώπιση αυτού του προβλήματος, παρουσιάζεται μια πρωτότυπη μεθοδολογία για ερμηνεύσιμη ανίχνευση ανωμαλιών χωρίς επίβλεψη, η οποία συνδυάζει την υψηλή απόδοση με την επεξηγησιμότητα. Η προτεινόμενη μέθοδος ενσωματώνει ροές κανονικοποίησης (normalizing flows) και δέντρα απόφασης (decision trees), συνδυάζοντας την ισχυρή ικανότητα γενεσιουργούς μοντέλοποίησης της βαθιάς μάθησης με τη διαφάνεια των δέντρων απόφασης. Οι ροές κανονικοποίησης χρησιμοποιούνται για την εκμάθηση της υποκείμενης κατανομής πιθανότητας μη ετικετοποιημένων δεδομένων και τη δημιουργία ψευδοετικετών για την ταξινόμηση των παρατηρήσεων. Στη συνέχεια, τα δέντρα απόφασης εφαρμόζουν τεχνικές επιβλεπόμενης μάθησης πάνω στις ψευδοετικέτες, παρέχοντας σαφείς κανόνες απόφασης και ιεράρχηση χαρακτηριστικών. Με αυτόν τον τρόπο, η μέθοδος προσφέρει αξιοποιήσιμες πληροφορίες για τους παράγοντες που οδηγούν στην ανίχνευση ανωμαλιών και ενισχύει την ερμηνευσιμότητα μέσω διαφανών εξηγήσεων.

Οι βασικές συνεισφορές αυτού του κεφαλαίου είναι οι ακόλουθες:

- Εισαγωγή μιας νέας μεθόδου που συνδυάζει ροές κανονικοποίησης και δέντρα απόφασης για βελτιωμένη ερμηνευσιμότητα στην ανίχνευση ανωμαλιών χωρίς επίβλεψη.
- Σύγκριση με 23 αλγορίθμους ανίχνευσης ανωμαλιών σε 17 σύνολα δεδομένων, με αξιολόγηση βάσει των μετρικών precision, recall, F1 score και Matthews correlation coefficient και στατιστική επικύρωση μέσω κατάταξης Friedman και post hoc ελέγχων.
- Προσδιορισμός των συνθηκών όπου η μέθοδος αποδίδει καλύτερα και των τομέων που χρειάζονται βελτίωση, με βάση την απόδοση σε διάφορα σύνολα δεδομένων.
- Χρήση ψευδοετικετών για την εφαρμογή επιβλεπόμενης μάθησης σε ένα πλαίσιο χωρίς επίβλεψη, παρέχοντας σαφείς εξηγήσεις μέσω δενδροειδών διαγραμμάτων, κανόνων απόφασης και ανάλυσης σπουδαιότητας χαρακτηριστικών.

Στη συνέχεια, αναλύονται τα βασικά μέρη της προτεινόμενης μεθόδου, ξεκινώντας από τα δέντρα απόφασης. Τα δέντρα απόφασης είναι ιδιαίτερα κατάλληλα για αυτόν τον σκοπό λόγω της εγγενούς διαφάνειάς τους, ως μοντέλα «λευκού κουτιού». Σε αντίθεση με λιγότερο ερμηνεύσιμα μοντέλα όπως το **SVM**, τα δέντρα απόφασης δημιουργούν εύκολα κατανοητούς κανόνες «αν-τότε». Επιπλέον, μπορούν να καταγράψουν μη γραμμικές εξαρτήσεις μεταξύ των μεταβλητών, ένας περιορισμός που υπάρχει σε άλλα ερμηνεύσιμα μοντέλα, όπως η λογιστική παλινδρόμηση, η οποία υποθέτει γραμμικότητα. Οι αναπαραστάσεις τους σε δενδροειδή διαγράμματα προσθέτουν ένα επιπλέον επίπεδο σαφήνειας, καθιστώντας τα ιδιαίτερα ωφέλιμα σε εφαρμογές όπου η επεξηγησιμότητα είναι υψίστης σημασίας. Επιπρόσθετα, ενώ η πρόβλεψη σε πραγματικό χρόνο δεν είναι ο πρωταρχικός στόχος αυτής της εργασίας, τα δέντρα απόφασης επιτρέπουν πολύ ταχύτερη συμπερασματολογία σε σύγκριση με μοντέλα όπως το **KNN**, κάτι που είναι πλεονεκτικό για αποφάσεις ευαίσθητες στον χρόνο. Αυτός ο συνδυασμός ερμηνευσιμότητας, η ικανότητα διαχείρισης μη γραμμικών σχέσεων και η υπολογιστική απόδοση καθιστούν τα δέντρα απόφασης ιδιαίτερα αποτελεσματικά σε πρακτικές εφαρμογές που απαιτούν γρήγορες και κατανοητές εξόδους μοντέλου.

Ένα άλλο κρίσιμο κομμάτι της προτεινόμενης μεθόδου είναι οι ροές κανονικοποίησης. Οι ροές κανονικοποίησης [274] είναι μια κατηγορία προηγμένων γενεσιουργών μοντέλων βαθιάς μάθησης που έχουν σχεδιαστεί για να μαθαίνουν περίπλοκες κατανομές δεδομένων μέσω μιας σειράς αντιστρέψιμων μετασχηματισμών. Η κεντρική αρχή των ροών κανονικοποίησης περιλαμβάνει τη μετατροπή μιας απλής βασικής κατανομής, όπως μια τυπική πολυμεταβλητή κανονική κατανομή, σε μια σύνθετη κατανομή στόχου που αντικατοπτρίζει τη δομή των δεδομένων. Αυτή η μετατροπή πραγματοποιείται μέσω μιας ακολουθίας αμφιμονοσήμαντων, διαφορίσιμων απεικονίσεων που επιτρέπουν έναν αποτελεσματικό υπολογισμό της συνάρτησης πυκνότητας πιθανότητας (**PDF**).

Σε επίπεδο μαθηματικής θεμελίωσης, για ένα σύνολο δεδομένων $\mathbf{x} \in \mathbb{R}^n$, ο στόχος είναι η εκμάθηση της κατανομής πιθανότητας $p_{\mathbf{x}}(\mathbf{x})$. Οι ροές κανονικοποίησης το επιτυγχάνουν εφαρμόζοντας μια ακολουθία αντιστρέψιμων

μετασχηματισμών $f_1, f_2, \ldots, f_K$ σε μια απλούστερη βασική κατανομή $p_z(\mathbf{z})$, όπου το $\mathbf{z} \in \mathbb{R}^n$ προέρχεται από μια τυπική κανονική κατανομή $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Αυτή η σειρά μετασχηματισμών απεικονίζει τη βασική κατανομή στην επιθυμητή κατανομή ως εξής:

$$\mathbf{x} = f_K \circ f_{K-1} \circ \cdots \circ f_1(\mathbf{z}) \tag{B.4}$$

Η πυκνότητα πιθανότητας για το $\mathbf{x}$ μπορεί στη συνέχεια να υπολογιστεί χρησιμοποιώντας τον τύπο αλλαγής μεταβλητών:

$$p_{\mathbf{x}}(\mathbf{x}) = p_z(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right|^{-1} \tag{B.5}$$

Αυτό μπορεί να εκφραστεί περαιτέρω χρησιμοποιώντας τον κανόνα της αλυσίδας ως εξής:

$$p_{\mathbf{x}}(\mathbf{x}) = p_z(\mathbf{z}) \prod_{k=1}^{K} \left| \det \left(\frac{\partial f_k}{\partial f_{k-1}} \right) \right|^{-1} \tag{B.6}$$

Σε αυτή τη διατύπωση, το $\left| \det \left(\frac{\partial f_k}{\partial f_{k-1}} \right) \right|$ αντιπροσωπεύει την ορίζουσα του Ιακωβιανού πίνακα για τον μετασχηματισμό f_k. Η αντιστρεψιμότητα των μετασχηματισμών καθιστά τον υπολογισμό της ορίζουσας αποτελεσματικό.

Σε πρακτικές εφαρμογές, οι υπολογισμοί απλοποιούνται χρησιμοποιώντας τον λογάριθμο της πυκνότητας πιθανότητας. Αυτή η προσέγγιση μετατρέπει το γινόμενο των οριζουσών σε άθροισμα, καθιστώντας τους υπολογισμούς πιο εφικτούς. Για πολυμεταβλητά δεδομένα, η λογαριθμική μορφή της πυκνότητας πιθανότητας εκφράζεται ως εξής:

$$\log p_{\mathbf{x}}(\mathbf{x}) = \log p_z(f^{-1}(\mathbf{x})) + \log \left| \det \frac{df^{-1}(\mathbf{x})}{d\mathbf{x}} \right| \tag{B.7}$$

Εδώ, το $f^{-1}(\mathbf{x})$ δηλώνει το αντίστροφο της σειράς μετασχηματισμών που εφαρμόζονται στο $\mathbf{x}$, και το $\frac{df^{-1}(\mathbf{x})}{d\mathbf{x}}$ είναι ο Ιακωβιανός του αντίστροφου μετασχηματισμού.

Η προτεινόμενη μέθοδος αξιολογήθηκε σε 17 σύνολα δεδομένων και συγκρίθηκε με 23 μοντέλα χρησιμοποιώντας ως μετρικές το F1 score, τον συντελεστή

συσχέτισης MCC, το precision και το recall. Η στατιστική ανάλυση έδειξε ότι δεν υπάρχουν στατιστικά σημαντικές διαφορές στην απόδοση μεταξύ της προτεινόμενης μεθόδου και των συγκρινόμενων μοντέλων για τρεις από τις τέσσερις μετρικές (precision, F1 score, MCC). Το τεστ κατάταξης Friedman και οι post hoc αναλύσεις υποστήριξαν αυτά τα αποτελέσματα. Παρά τη συγκρίσιμη απόδοση, η προτεινόμενη μέθοδος υπερέχει σημαντικά σε επίπεδο ερμηνευσιμότητας σε σχέση με τα ανταγωνιστικά μοντέλα.

Η μέθοδος επιδεικνύει ανταγωνιστική απόδοση στις διάφορες μετρικές, με ιδιαίτερη υπεροχή στο precision. Η ικανότητα ακριβούς αναγνώρισης των πραγματικών ανωμαλιών, ελαχιστοποιώντας παράλληλα τους «ψευδείς συναγερμούς», την καθιστά ιδανική για εφαρμογές όπου η μείωση των ψευδώς θετικών προβλέψεων είναι κρίσιμη. Αν και δεν κατατάσσεται στην κορυφή όσον αφορά το F1 score ή το MCC, η απόδοσή της παραμένει σταθερά συγκρίσιμη με κορυφαία μοντέλα όπως το IForest και το OCSVM, χωρίς στατιστικά σημαντικές διαφορές.

Ένα βασικό πλεονέκτημα της μεθόδου είναι η ερμηνευσιμότητά της. Η χρήση δέντρων απόφασης παράγει σαφείς και κατανοητούς κανόνες, επιτρέποντας την κατανόηση των παραγόντων που οδηγούν στην ανίχνευση ανωμαλιών. Επιπλέον, η αντιστρεψιμότητα των ροών κανονικοποίησης παρέχει πρόσθετες πληροφορίες για τους μετασχηματισμούς των δεδομένων, ενισχύοντας τη διαφάνεια. Αυτά τα χαρακτηριστικά είναι ιδιαίτερα πολύτιμα σε τομείς όπου οι αποφάσεις πρέπει να είναι διαφανείς και αιτιολογημένες.

Ωστόσο, η μέθοδος παρουσιάζει περιορισμούς στο recall, καθώς δεν καταφέρνει να εντοπίσει όλες τις πραγματικές ανωμαλίες σε ορισμένα σύνολα δεδομένων. Αυτός ο συμβιβασμός μεταξύ precision και recall είναι πιο έντονος σε σενάρια όπου η ελαχιστοποίηση των ψευδώς θετικών αποτελεσμάτων είναι προτεραιότητα, ενδεχομένως εις βάρος της ανίχνευσης όλων των ανωμαλιών. Μελλοντική έρευνα θα μπορούσε να επικεντρωθεί στη βελτιστοποίηση της ισορροπίας μεταξύ αυτών των μετρικών, βελτιώνοντας την ικανότητα ανίχνευσης ανωμαλιών χωρίς να θίγεται το precision. Κατά την αξιολόγηση της υπολογιστικής απόδοσης, γίνεται διάκριση μεταξύ της εκπαίδευσης, που συνήθως πραγ-

ματοποιείται «offline» και απαιτεί περισσότερο χρόνο, και της βαθμολόγησης, που είναι κρίσιμης σημασίας για εφαρμογές σε πραγματικό χρόνο.

Συνοψίζοντας την αξιολόγηση, η προτεινόμενη μέθοδος επιδεικνύει ανταγωνιστική απόδοση στην ανίχνευση ανωμαλιών και υπερέχει στην ερμηνευσιμότητα σε σύγκριση με άλλα μοντέλα. Η έμφαση στο precision την καθιστά ιδιαίτερα χρήσιμη σε περιπτώσεις όπου η αποφυγή ψευδών συναγερμών είναι πρωταρχικής σημασίας. Παρά τους περιορισμούς στο recall σε ορισμένα σύνολα δεδομένων, η δυνατότητα ερμηνείας των αποτελεσμάτων και η ακρίβεια στην ανίχνευση καθιστούν την προσέγγιση αυτή αξιόλογη. Η διάκριση μεταξύ του χρόνου εκπαίδευσης και της ταχύτητας βαθμολόγησης είναι σημαντική για την πρακτική εφαρμογή της μεθόδου. Περαιτέρω, η ερμηνευσιμότητα που παρέχεται από τα δέντρα απόφασης και τις ροές κανονικοποίησης είναι κρίσιμη για την κατανόηση των υποκείμενων λόγων πίσω από την ταξινόμηση μιας παρατήρησης ως ανώμαλης. Οι κανόνες «αν-τότε» των δέντρων απόφασης προσφέρουν άμεση εικόνα για τις συνθήκες που οδηγούν σε μια ανίχνευση, ενώ η αντιστρεψιμότητα των ροών κανονικοποίησης επιτρέπει την ανάλυση των μετασχηματισμών που υφίστανται τα δεδομένα. Αυτή η διαφάνεια είναι απαραίτητη σε ευαίσθητους τομείς όπου οι αποφάσεις πρέπει να είναι όχι μόνο ακριβείς αλλά και κατανοητές και δικαιολογημένες.

Επισκόπηση των Συνεισφορών της Διατριβής

Η παρούσα διατριβή συνέβαλε στην ανίχνευση ανωμαλιών χωρίς επίβλεψη, εστιάζοντας στην αντιμετώπιση των προκλήσεων που αφορούν 1. τη διαδικασία επιλογής του κατάλληλου μοντέλου με στόχο τη δημιουργία ισχυρών ομάδων ταξινομητών, και 2. την ερμηνεία των αποτελεσμάτων.

Στο κεφάλαιο που επικεντρώνεται στη μετα-μάθηση, αναπτύχθηκε μια νέα μέθοδος για τη δημιουργία ισχυρών ομάδων ταξινομητών μέσω τεχνικών φασματικής ανάλυσης για επιλογή χαρακτηριστικών χωρίς επίβλεψη. Σε αντίθεση με απλούστερες τεχνικές που συνδυάζουν αδιακρίτως τις προβλέψεις πολλών ανιχνευτών, η προτεινόμενη προσέγγιση προχωρά σε μια ενδελεχή αξιολόγηση και επιλογή των ανιχνευτών που αποδεικνύονται πιο συναφείς με τη δομή του

εκάστοτε συνόλου δεδομένων. Η στοχευμένη αυτή διαδικασία επιλογής συμβάλλει σημαντικά τόσο στην ακρίβεια του εντοπισμού των ανωμαλιών όσο και στη συνολική αποτελεσματικότητα των συστημάτων ανίχνευσης. Κατ΄ αυτόν τον τρόπο, επιτυγχάνεται μια πιο αξιόπιστη και ανθεκτική διαδικασία επιλογής του βέλτιστου μοντέλου, γεγονός ιδιαίτερα κρίσιμο σε περιβάλλοντα όπου η ακρίβεια και η ανθεκτικότητα των αποτελεσμάτων είναι υψίστης προτεραιότητας. Η πρακτική χρησιμότητα και η αποδεδειγμένη αποτελεσματικότητα της προτεινόμενης μεθόδου μετα-μάθησης επιβεβαιώθηκε μέσα από εκτεταμένες δοκιμές σε διάφορα σύνολα δεδομένων, κυρίως σε μορφή πινάκων.

Στο κεφάλαιο που είναι αφιερωμένο στην ταξινομία των μεθόδων ερμηνευσιμότητας στη μηχανική μάθηση, αναπτύχθηκε ένα πλήρες και συστηματικό πλαίσιο ταξινόμησης. Η ταξινομία αυτή παρέχει μια καλά οργανωμένη δομή για την κατηγοριοποίηση των διαφόρων προσεγγίσεων ερμηνευσιμότητας, με βάση βασικές διαστάσεις όπως το αν η ερμηνεία αφορά μεμονωμένες προβλέψεις ή το μοντέλο συνολικά (local vs. global), αν οι τεχνικές είναι ανεξάρτητες από το συγκεκριμένο μοντέλο ή προσαρμοσμένες σε αυτό (model-agnostic vs. model-specific), και ανάλογα με τον τύπο των δεδομένων στον οποίο μπορούν να εφαρμοστούν. Μέσα από την ανάδειξη των αλληλεπιδράσεων και των αντισταθμίσεων μεταξύ των διαφόρων μεθόδων, η ταξινομία αυτή προσφέρει ουσιαστική καθοδήγηση σε ερευνητές και επαγγελματίες, βοηθώντας τους να επιλέξουν τις στρατηγικές ερμηνευσιμότητας που ανταποκρίνονται καλύτερα στους ειδικούς στόχους και τους περιορισμούς τους. Ο απώτερος σκοπός αυτού του πλαισίου είναι να διευκολύνει την ουσιαστική ενσωμάτωση της ερμηνευσιμότητας στις διαδικασίες της μηχανικής μάθησης, ιδίως σε εκείνους τους τομείς όπου η διαφάνεια και η δυνατότητα λογοδοσίας αποτελούν θεμελιώδεις απαιτήσεις.

Στο τρίτο κεφάλαιο, εισήχθη μια πρωτότυπη προσέγγιση για την ανίχνευση ανωμαλιών και επιτεύχθηκε η συνδυαστική χρήση ροών κανονικοποίησης και δέντρων απόφασης. Η συνδυαστική αυτή μέθοδος επιτυγχάνει μια αξιοσημείωτη ισορροπία ανάμεσα στην ερμηνευσιμότητα των αποτελεσμάτων και στην υψηλή ακρίβεια της ανίχνευσης. Παρέχει σαφείς και εύληπτους κανόνες λήψης

αποφάσεων, διατηρώντας παράλληλα εξαιρετικές επιδόσεις στην αναγνώριση των ανώμαλων παρατηρήσεων. Η προσέγγιση αυτή αναδεικνύεται ιδιαίτερα χρήσιμη σε βιομηχανίες και τομείς όπου η διαφανής διαδικασία λήψης αποφάσεων είναι καθοριστικής σημασίας για την καλλιέργεια εμπιστοσύνης και τη διασφάλιση της αξιοπιστίας των συστημάτων (π.χ. υγεία, τράπεζες, αυτο-οδηγούμενα οχήματα). Επιτρέπει τόσο την ακριβή ανίχνευση των ανωμαλιών όσο και την παροχή σαφών εξηγήσεων σχετικά με τους υποκείμενους λόγους που οδήγησαν στη συγκεκριμένη ταξινόμηση. Η αποτελεσματικότητα της προτεινόμενης μεθόδου τεκμηριώθηκε μέσα από πειράματα σε δεδομένα διαφόρων μορφών, συμπεριλαμβανομένων πινάκων και εικόνων, τα οποία καταδεικνύουν τη δυνατότητά της να ενισχύσει την ερμηνευσιμότητα χωρίς να υπονομεύει την ακρίβεια της ανίχνευσης.

Περιορισμοί και Μελλοντικές Κατευθύνσεις

Παρόλο που η παρούσα διατριβή αντιμετώπισε σημαντικές προκλήσεις στον τομέα της ανίχνευσης ανωμαλιών χωρίς επίβλεψη, υπάρχουν εγγενείς περιορισμοί και σημαντικές ευκαιρίες για μελλοντική έρευνα που παραμένουν ανοικτές.

Ένας περιορισμός του αλγορίθμου μετα-μάθησης που παρουσιάστηκε είναι ότι δεν εξετάστηκαν πλήρως η επέκτασή του σε δεδομένα πολύ υψηλών διαστάσεων και η απόδοσή του σε εφαρμογές πραγματικού χρόνου. Επιπλέον, η εφαρμογή του σε μη δομημένα δεδομένα και η ανθεκτικότητά του στην εννοιολογική μετατόπιση (concept drift) αποτελούν πεδία που χρήζουν περαιτέρω εξέτασης. Ως εκ τούτου, μελλοντικές ερευνητικές κατευθύνσεις περιλαμβάνουν:

- Τη διερεύνηση της ικανότητας του πλαισίου να κλιμακώνεται αποτελεσματικά σε σύνολα δεδομένων υψηλών διαστάσεων και σε εφαρμογές πραγματικού χρόνου, διασφαλίζοντας την πρακτική του αξία σε περιβάλλοντα μεγάλου όγκου δεδομένων.
- Την προσαρμογή του πλαισίου για την ανάλυση μη δομημένων δεδομένων (π.χ., χρονοσειρές, γράφοι, κείμενο), ενσωματώνοντας εξειδικευμένους μηχανισμούς για τις εγγενείς δομικές ή χρονικές τους ιδιότητες.

- Την ενσωμάτωση μηχανισμών για την αντιμετώπιση του φαινομένου της εννοιολογικής μετατόπισης, όπου η κατανομή των δεδομένων εξελίσσεται, ώστε να ενισχυθεί η προσαρμοστικότητα του αλγορίθμου σε δυναμικά περιβάλλοντα.

Η ταξινομία μεθόδων ερμηνευσιμότητας που αναπτύχθηκε, αν και χρήσιμη, εστιάζει κυρίως σε καθιερωμένα πλαίσια. Ένας περιορισμός είναι ότι δεν καλύπτει πλήρως τις ιδιαίτερες προκλήσεις ερμηνείας που ανακύπτουν στη μάθηση χωρίς επίβλεψη ή σε σενάρια με πολλαπλούς τύπους δεδομένων (πολυτροπικά - multimodal). Μελλοντική έρευνα θα μπορούσε να εμπλουτίσει την ταξινομία, εστιάζοντας ιδιαίτερα σε αυτές τις προκλήσεις. Η διεύρυνσή της ώστε να συμπεριλάβει αυτές τις περιοχές θα την καταστήσει πιο ολοκληρωμένη. Επιπλέον, η ενσωμάτωση πρακτικών οδηγιών και βέλτιστων πρακτικών θα την κάνει συμβατή με τις σύγχρονες ανάγκες ερευνητών και επαγγελματιών. Αυτό θα συμβάλει στην ανάπτυξη πιο διαφανών και αξιόπιστων συστημάτων μηχανικής μάθησης.

Η προτεινόμενη μέθοδος ανίχνευσης ανωμαλιών που συνδυάζει ροές κανονικοποίησης και δέντρα απόφασης παρουσιάζει επίσης ευκαιρίες για περαιτέρω διερεύνηση. Ένας τρέχων περιορισμός είναι ο παρατηρούμενος συμβιβασμός μεταξύ precision και recall, κάτι που απαιτεί βελτίωση σε εφαρμογές όπου το υψηλό recall είναι κρίσιμο. Επιπλέον, η υπολογιστική πολυπλοκότητα της εκπαίδευσης, ιδιαίτερα για μεγάλα σύνολα δεδομένων, αποτελεί πρόκληση. Μελλοντικές κατευθύνσεις για την ενίσχυση της μεθόδου περιλαμβάνουν:

- Τη βελτίωση του συμβιβασμού precision / recall, πιθανώς μέσω τροποποιήσεων στην αρχιτεκτονική των ροών κανονικοποίησης ή με την εξέταση εναλλακτικών τεχνικών εκτίμησης πυκνότητας πιθανότητας.
- Τη διερεύνηση τρόπων μείωσης της υπολογιστικής πολυπλοκότητας, όπως η απλοποίηση της αρχιτεκτονικής ή η χρήση εναλλακτικών προσεγγίσεων εκπαίδευσης, για τη βελτίωση της αποδοτικότητας και της επεκτασιμότητας.
- Την επέκταση της εφαρμοσιμότητάς της σε άλλους τύπους δεδομένων, όπως χρονοσειρές ή δεδομένα γραφημάτων.
- Τη διερεύνηση της ικανότητάς της να διαχειρίζεται αποτελεσματικά την

εννοιολογική μετατόπιση.

Συνοψίζοντας, η παρούσα διατριβή εισήγαγε μεθόδους που στοχεύουν στην αντιμετώπιση συγκεκριμένων προβλημάτων στην ανίχνευση ανωμαλιών χωρίς επίβλεψη και στην ερμηνευσιμότητα των μοντέλων. Η έρευνα ανέδειξε κοινές, οριζόντιες προκλήσεις και ευκαιρίες για το μέλλον. Κεντρικοί άξονες για μελλοντική εργασία περιλαμβάνουν την κλιμάκωση των μεθόδων για πολύ μεγάλα, υψηλών διαστάσεων σύνολα δεδομένων, την προσαρμογή τους σε ποικίλους τύπους δεδομένων (χρονοσειρές, γράφοι, πολυτροπικά δεδομένα), και τη βελτίωση της υπολογιστικής απόδοσης για υποστήριξη εφαρμογών πραγματικού χρόνου. Εξίσου σημαντική είναι η αντιμετώπιση της εννοιολογικής μετατόπισης για ανθεκτικότητα σε δυναμικά περιβάλλοντα και η περαιτέρω βελτίωση της ισορροπίας μεταξύ precision και recall. Η αντιμετώπιση αυτών των ζητημάτων θα επιτρέψει στις αναπτυγμένες μεθόδους να αποκτήσουν ευρύτερη εφαρμογή, συμβάλλοντας στην πρόοδο προς πιο ανθεκτικά, διαφανή και προσαρμόσιμα συστήματα μηχανικής μάθησης.

Bibliography

[1] Rémi Domingues, Maurizio Filippone, Pietro Michiardi, and Jihane Zouaoui. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern recognition*, 74:406–421, 2018. [1, B]

[2] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004. [1, 1.1, B]

[3] Tehreem Ashfaq, Rabiya Khalid, Adamu Sani Yahaya, Sheraz Aslam, Ahmad Taher Azar, Safa Alsafari, and Ibrahim A Hameed. A machine learning and blockchain based efficient fraud detection mechanism. *Sensors*, 22(19):7162, 2022. [1, B]

[4] Waleed Hilal, S Andrew Gadsden, and John Yawney. Financial fraud: a review of anomaly detection techniques and recent advances. *Expert systems With applications*, 193:116429, 2022. [1, B]

[5] Abdellatif Bouzid-Daho and Patrick Siarry. Detection, identification and tracking of anomaly in medical images. In *2023 IEEE International Conference on Advances in Data-Driven Analytics And Intelligent Systems (ADACIS)*, pages 1–4. IEEE, 2023. [1, B]

[6] Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Deep learning for medical anomaly detection–a survey. *ACM Computing Surveys (CSUR)*, 54(7):1–37, 2021. [1, B]

[7] Huimin Zhao, Yongshun Gao, and Wu Deng. Defect detection using shuffle net-ca-ssd lightweight network for turbine blades in iot. *IEEE Internet of Things Journal*, 2024. [1, B]

[8] Weihan Li, Dunke Liu, Yang Li, Ming Hou, Jie Liu, Zhen Zhao, Aibin Guo,

Huimin Zhao, and Wu Deng. Fault diagnosis using variational autoencoder gan and focal loss cnn under unbalanced data. *Structural Health Monitoring*, page 14759217241254121, 2024. [1, B]

[9] Charu C Aggarwal and S Yu Philip. An effective and efficient algorithm for high-dimensional outlier detection. *The VLDB journal*, 14(2):211–221, 2005. [1, B]

[10] Omar Alghushairy, Raed Alsini, Terence Soule, and Xiaogang Ma. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing*, 5(1):1, 2021. [1, B]

[11] Xin Luna Dong and Divesh Srivastava. Big data integration. In *2013 IEEE 29th international conference on data engineering (ICDE)*, pages 1245–1248. IEEE, 2013. [1, B]

[12] Hancong Liu, Sirish Shah, and Wei Jiang. On-line outlier detection and data cleaning. *Computers & chemical engineering*, 28(9):1635–1647, 2004. [1, B]

[13] Charu C Aggarwal. An introduction to outlier analysis. In *Outlier analysis*, pages 1–34. Springer, 2017. [1, 2.2, B, B]

[14] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009. [1, B]

[15] Hongzhi Wang, Mohamed Jaward Bah, and Mohamed Hammad. Progress in outlier detection techniques: A survey. *IEEE Access*, 7:107964–108000, 2019. [1, B]

[16] Fanrong Meng, Guan Yuan, Shaoqian Lv, Zhixiao Wang, and Shixiong Xia. An overview on trajectory outlier detection. *Artificial Intelligence Review*, 52(4):2437–2456, 2019. [1.1, B]

[17] Charu C Aggarwal and Charu C Aggarwal. Supervised outlier detection. *Outlier Analysis*, pages 219–248, 2017. [1.1.1]

[18] Shohei Hido, Yuta Tsuboi, Hisashi Kashima, Masashi Sugiyama, and Takafumi Kanamori. Statistical outlier detection using direct density ratio estimation. *Knowledge and information systems*, 26:309–336, 2011. [1.1.1]

[19] Miryam Elizabeth Villa-Pérez, Miguel A Alvarez-Carmona, Octavio Loyola-Gonzalez, Miguel Angel Medina-Pérez, Juan Carlos Velazco-Rossell, and Kim-Kwang Raymond Choo. Semi-supervised anomaly detection algorithms: A comparative summary and future research directions. *Knowledge-Based Systems*, 218:106878, 2021. [1.1.2]

[20] NNR Murty Ranga Suri, M Narasimha Murty, and G Athithan. *Outlier detection: techniques and applications*. Springer, 2019. [1.1.3]

[21] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012. [1.1.3]

[22] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000. [1.2, 2.2, 2.2.1, 4.3.2.4]

[23] Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track*, 1:59–63, 2012. [1.2, 2.2.1, 4.3.2.4]

[24] Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 Proceedings 6*, pages 535–548. Springer, 2002. [1.2]

[25] Edwin M Knorr, Raymond T Ng, and Vladimir Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3):237–253, 2000. [1.2]

[26] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 427–438, 2000. [1.2, 2.2, 2.2.1, 4.3.2.4]

[27] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452, 2008. [1.2, 2.2.1, 4.3.2.4]

[28] Alberto Barbado, Óscar Corcho, and Richard Benjamins. Rule extraction in unsupervised anomaly detection for model explainability: Application to oneclass svm. *Expert Systems with Applications*, 189:116100, 2022. [1.2, 4, 4.1.1, B]

[29] Ruoyu Li, Qing Li, Yu Zhang, Dan Zhao, Yong Jiang, and Yong Yang. Interpreting unsupervised anomaly detection in security via rule extraction. *Advances in Neural Information Processing Systems*, 36, 2024. [1.2, 4, B]

[30] Vasilis Papastefanopoulos, Pantelis Linardatos, and Sotiris Kotsiantis. Unsupervised outlier detection: a meta-learning algorithm based on feature selection. *Electronics*, 10(18):2236, 2021. [1.3]

[31] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020. [1.3]

[32] Vasilis Papastefanopoulos, Pantelis Linardatos, and Sotiris Kotsiantis. Combining normalizing flows with decision trees for interpretable unsupervised outlier detection. *Engineering Applications of Artificial Intelligence*, 141:109770, 2025. [1.3]

[33] Charu C Aggarwal. Outlier ensembles: position paper. *ACM SIGKDD Explorations Newsletter*, 14(2):49–58, 2013. [2.1]

[34] Arthur Zimek, Ricardo JGB Campello, and Jørg Sander. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *Acm Sigkdd Explorations Newsletter*, 15(1):11–22, 2014. [2.1]

[35] Akanksha Mukhriya and Rajeev Kumar. Exploring ensembles for unsupervised outlier detection: an empirical analysis. *Towards Extensible and Adaptable Methods in Computing*, pages 225–237, 2018. [2.1]

[36] Henrique O Marques, Ricardo JGB Campello, Jørg Sander, and Arthur

Zimek. Internal evaluation of unsupervised outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(4):1–42, 2020. [2.1]

[37] Stamatios-Aggelos N Alexandropoulos, Sotiris B Kotsiantis, Violetta E Piperigou, and Michael N Vrahatis. A new ensemble method for outlier identification. In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 769–774. IEEE, 2020. [2.1]

[38] Yue Zhao, Ryan Rossi, and Leman Akoglu. Automatic unsupervised outlier model selection. *Advances in Neural Information Processing Systems*, 34:4489–4502, 2021. [2.1]

[39] Xiaoyong Li, Yong Zhang, Huimin Cheng, Feifei Zhou, and Baocai Yin. An unsupervised ensemble clustering approach for the analysis of student behavioral patterns. *Ieee Access*, 9:7076–7091, 2021. [2.1]

[40] Siddharth Chaurasia, Sagar Goyal, and Manish Rajput. Outlier detection using autoencoder ensembles: A robust unsupervised approach. In *2020 International Conference on Contemporary Computing and Applications (IC3A)*, pages 76–80. IEEE, 2020. [2.1]

[41] Boya Ouyang, Yu Song, Yuhai Li, Gaurav Sant, and Mathieu Bauchy. Ebod: An ensemble-based outlier deguide to deep learning in healthcaretection algorithm for noisy datasets. *Knowledge-Based Systems*, 231:107400, 2021. [2.1]

[42] Hamed Sarvari, Carlotta Domeniconi, Bardh Prenkaj, and Giovanni Stilo. Unsupervised boosting-based autoencoder ensembles for outlier detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 91–103. Springer, 2021. [2.1]

[43] Bodhan Chakraborty, Agneet Chaterjee, Samir Malakar, and Ram Sarkar. An iterative approach to unsupervised outlier detection using ensemble method and distance-based data filtering. *Complex & Intelligent Systems*, 8(4):3215–3230, 2022. [2.1]

[44] Li Cheng, Yijie Wang, Xinwang Liu, and Bin Li. Outlier detection ensemble with embedded feature selection. In *Proceedings of the AAAI Conference on*

Artificial Intelligence, volume 34, pages 3503–3512, 2020. [2.1]

[45] Yue Zhao, Xiyang Hu, Cheng Cheng, Cong Wang, Cao Xiao, Yunlong Wang, Jimeng Sun, and Leman Akoglu. Suod: A scalable unsupervised outlier detection framework. *arXiv preprint arXiv:2003.05731*, 2020. [2.1]

[46] Akanksha Mukhriya and Rajeev Kumar. Building outlier detection ensembles by selective parameterization of heterogeneous methods. *Pattern Recognition Letters*, 146:126–133, 2021. [2.1]

[47] Nicholas Merrill and Colin C Olson. Unsupervised ensemble-kernel principal component analysis for hyperspectral anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 112–113, 2020. [2.1]

[48] Lynda Boukela, Gongxuan Zhang, Samia Bouzefrane, and Junlong Zhou. An outlier ensemble for unsupervised anomaly detection in honeypots data. *Intelligent Data Analysis*, 24(4):743–758, 2020. [2.1]

[49] Duc C Le and Nur Zincir-Heywood. Anomaly detection for insider threats using unsupervised ensembles. *IEEE Transactions on Network and Service Management*, 18(2):1152–1164, 2021. [2.1]

[50] Peng An, Zhiyuan Wang, and Chunjiong Zhang. Ensemble unsupervised autoencoders and gaussian mixture model for cyberattack detection. *Information Processing & Management*, 59(2):102844, 2022. [2.1]

[51] Spyridon Plakias and Yiannis S Boutalis. A novel information processing method based on an ensemble of auto-encoders for unsupervised fault detection. *Computers in Industry*, 142:103743, 2022. [2.1]

[52] Alexander Vosseler. Unsupervised insurance fraud prediction based on anomaly detector ensembles. *Risks*, 10(7):132, 2022. [2.1]

[53] Wasim Khan and Mohammad Haroon. An unsupervised deep learning ensemble model for anomaly detection in static attributed social networks. *International Journal of Cognitive Computing in Engineering*, 3:153–160, 2022. [2.1]

[54] Roy Thomas and JE Judith. Voting-based ensemble of unsupervised outlier

detectors. In *Advances in Communication Systems and Networks: Select Proceedings of ComNet 2019*, pages 501–511. Springer, 2020. [2.1]

[55] Biswarup Ray, Soulib Ghosh, Shameem Ahmed, Ram Sarkar, and Mita Nasipuri. Outlier detection using an ensemble of clustering algorithms. *Multimedia Tools and Applications*, 81(2):2681–2709, 2022. [2.1]

[56] Akash Saha, Agneet Chatterjee, Soulib Ghosh, Neeraj Kumar, and Ram Sarkar. An ensemble approach to outlier detection using some conventional clustering algorithms. *Multimedia Tools and Applications*, 80:35145–35169, 2021. [2.1]

[57] Jaehong Yu and Jihoon Kang. Clustering ensemble-based novelty score for outlier detection. *Engineering Applications of Artificial Intelligence*, 121:106164, 2023. [2.1]

[58] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George H Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12181–12193, 2022. [2.1, 4.3.2.4]

[59] Qing Cheng, Yun Zhou, Yanghe Feng, and Zhong Liu. An unsupervised ensemble framework for node anomaly behavior detection in social network. *Soft Computing*, 24(9):6421–6431, 2020. [2.1]

[60] Jiawei Yang, Sylwan Rahardja, and Susanto Rahardja. Regional ensemble for improving unsupervised outlier detectors. *IEEE Transactions on Artificial Intelligence*, 2024. [2.1]

[61] David Campos, Tung Kieu, Chenjuan Guo, Feiteng Huang, Kai Zheng, Bin Yang, and Christian S Jensen. Unsupervised time series outlier detection with diversity-driven convolutional ensembles. *Proceedings of the VLDB Endowment*, 15(3):611–623, 2021. [2.1]

[62] Iurii Katser, Viacheslav Kozitsin, Victor Lobachev, and Ivan Maksimov. Unsupervised offline changepoint detection ensembles. *Applied sciences*, 11(9):4280, 2021. [2.1]

[63] Abdul Jabbar Siddiqui and Azzedine Boukerche. Adaptive ensembles of

autoencoders for unsupervised iot network intrusion detection. *Computing*, 103(6):1209–1232, 2021. [2.1]

[64] Sevvandi Kandanaarachchi. Unsupervised anomaly detection ensembles using item response theory. *Information Sciences*, 587:142–163, 2022. [2.1]

[65] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern recognition letters*, 24(9-10):1641–1650, 2003. [2.2, 2.2.1, 4.3.2.4]

[66] Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1151–1157, 2007. [2.2, 2.2.2, 2.2.2.5, B]

[67] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997. [2.2]

[68] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001. [2.2.1, 4.3.2.4]

[69] Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999. [2.2.1, 4.3.2.4]

[70] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993. [2.2.1, 4.3.2.4]

[71] Aleksandar Lazarevic and Vipin Kumar. Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166, 2005. [2.2.1]

[72] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008. [2.2.1, 4.3.2.4]

[73] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. *Advances in neural information processing systems*, 18, 2005. [2.2.2]

[74] Salem Alelyani, Jiliang Tang, and Huan Liu. Feature selection for clustering: A review. *Data Clustering*, pages 29–60, 2018. [2.2.2]

[75] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. [2.2.2.5]

[76] Daniela M Witten and Robert Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010. [2.2.2.5]

[77] Joshua Zhexue Huang, Michael K Ng, Hongqiang Rong, and Zichen Li. Automated variable weighting in k-means type clustering. *IEEE transactions on pattern analysis and machine intelligence*, 27(5):657–668, 2005. [2.2.2.5]

[78] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003. [2.2.2.5]

[79] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of machine learning research*, 20(96):1–7, 2019. [2.3.1, 2.3.2.6, 4.2.4, 4.3.1, 4.3.2.4, 4.3.2.6]

[80] Amir E Khandani, Adlar J Kim, and Andrew W Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787, 2010. [3]

[81] Hong Hanh Le and Jean-Laurent Viviani. Predicting bank failure: An improvement by implementing a machine-learning approach to classical financial ratios. *Research in international business and finance*, 44:16–25, 2018. [3]

[82] Sumeet Dua, U Rajendra Acharya, and Prerna Dua. *Machine learning in healthcare informatics*, volume 56. Springer, 2014. [3]

[83] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019. [3]

[84] Alison Callahan and Nigam H Shah. Machine learning in healthcare. In *Key advances in clinical informatics*, pages 279–291. Elsevier, 2017. [3]

[85] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. [3, B]

[86] A Liaw. Classification and regression by randomforest. *R news*, 2002. [3, B]

[87] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016. [3, 4.4.7.7, B]

[88] S Weisberg. Applied linear regression, 2005. [3, B]

[89] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991. [3, B]

[90] David Gunning and David Aha. Darpa's explainable artificial intelligence (xai) program. *AI magazine*, 40(2):44–58, 2019. [3, B]

[91] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018. [3.1.1, 3.1.3, B]

[92] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. [3.1.1, 3.1.2, 3.1.3, B]

[93] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018. [3.1.1, 3.1.3, B]

[94] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018. [3.1.1, 3.1.3, B]

[95] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019. [3.1.1, B]

[96] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black

box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018. [3.1.3, B]

[97] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. [3.1.3, B]

[98] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020. [3.1.3, B]

[99] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. [3.2.1.1, 3.1]

[100] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. [3.2.1.1, 3.1]

[101] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. [3.2.1.1, 3.1]

[102] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR, 2017. [3.2.1.1]

[103] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*, pages 63–71. Springer, 2016. [3.2.1.1]

[104] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017. [3.2.1.1, 3.1]

[105] MD Zeiler. Visualizing and understanding convolutional networks. In *European conference on computer vision/arXiv*, volume 1311, 2014. [3.2.1.1, 3.1, 3.3.1, 3.4, B]

[106] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. [3.2.1.1]

[107] Pieter-Jan Kindermans, Kristof T Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. Learning how to explain neural networks: Patternnet and patternattribution. In *International Conference on Learning Representations*, 2018. [3.2.1.1, 3.1]

[108] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. [3.2.1.1, 3.1, 3.3.1]

[109] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. [3.2.1.1, 3.1, 3.3.1, 3.4, B]

[110] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. [3.2.1.1, 3.1]

[111] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert

Müller, Sven Dähne, and Pieter-Jan Kindermans. innvestigate neural networks! *Journal of machine learning research*, 20(93):1–8, 2019. [3.2.1.1]

[112] J Springenberg, Alexey Dosovitskiy, Thomas Brox, and M Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015. [3.1]

[113] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015. [3.1]

[114] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. [3.1]

[115] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016. [3.1]

[116] Sandy H. Huang, Nicolas Papernot, Ian J. Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. [3.1, 3.2.4.2]

[117] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. [3.1]

[118] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: randomized input sampling for explanation of black-box models. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 151. BMVA Press, 2018. [3.1]

[119] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i

trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. [3.2.1.2, 3.2, 3.3.1, 3.4, B]

[120] Damien Garreau and Ulrike Luxburg. Explaining the explainer: A first theoretical analysis of lime. In *International conference on artificial intelligence and statistics*, pages 1287–1296. PMLR, 2020. [3.2.1.2]

[121] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. [3.2.1.2, 3.2, 3.3.1, B]

[122] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics*, 24(1):44–65, 2015. [3.2.1.2, 3.2]

[123] Scott Lundberg. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017. [3.2.1.2, 3.2, 3.3.1, 3.4, B]

[124] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. [3.2.1.2, 3.2]

[125] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, 31, 2018. [3.2.1.2, 3.2]

[126] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017. [3.2.1.2, 3.2]

[127] Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4):1059–1086, 2020. [3.2.1.2, 3.2]

[128] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permu-

tation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010. [3.2]

[129] Mateusz Staniak and Przemyslaw Biecek. Explanations of model predictions with live and breakdown packages. *arXiv preprint arXiv:1804.01955*, 2018. [3.2]

[130] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International conference on machine learning*, pages 883–892. PMLR, 2018. [3.2]

[131] Muhammad Rehman Zafar and Naimul Mefraz Khan. Dlime: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *arXiv preprint arXiv:1906.10263*, 2019. [3.2]

[132] Karthik S Gurumoorthy, Amit Dhurandhar, Guillermo Cecchi, and Charu Aggarwal. Efficient data representation by selecting prototypes with importance weights. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 260–269. IEEE, 2019. [3.2]

[133] Ronny Luss, Pin-Yu Chen, Amit Dhurandhar, Prasanna Sattigeri, Karthik Shanmugam, and Chun-Chen Tu. Generating contrastive explanations with monotonic attribute functions. *arXiv preprint arXiv:1905.12698*, 3, 2019. [3.2]

[134] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 650–665. Springer, 2021. [3.2]

[135] Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102:349–391, 2016. [3.2.2, 3.3]

[136] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM*

SIGKDD international conference on knowledge discovery and data mining, pages 1721–1730, 2015. [3.2.2, 3.3, 3.3.2, B]

[137] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631, 2013. [3.2.2]

[138] Sanjeeb Dash, Oktay Gunluk, and Dennis Wei. Boolean decision rules via column generation. *Advances in neural information processing systems*, 31, 2018. [3.2.2, 3.3]

[139] Dennis Wei, Sanjeeb Dash, Tian Gao, and Oktay Gunluk. Generalized linear rule models. In *International conference on machine learning*, pages 6687–6696. PMLR, 2019. [3.2.2, 3.3]

[140] Peter McCullagh. *Generalized linear models*. Routledge, 2019. [3.2.2]

[141] Michael Hind, Dennis Wei, Murray Campbell, Noel CF Codella, Amit Dhurandhar, Aleksandra Mojsilović, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Ted: Teaching ai to explain its decisions. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 123–129, 2019. [3.2.2, 3.3]

[142] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015. [3.2.3, 3.4]

[143] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017. [3.2.3, 3.4]

[144] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016. [3.2.3, 3.4]

[145] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classifi-

cation without discrimination. *Knowledge and information systems*, 33(1):1–33, 2012. [3.2.3, 3.4]

[146] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data mining and knowledge discovery*, 21:277–292, 2010. [3.2.3, 3.4]

[147] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE international conference on data mining workshops*, pages 13–18. IEEE, 2009. [3.2.3, 3.4]

[148] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30, 2017. [3.2.3, 3.4]

[149] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018. [3.2.3, 3.4]

[150] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *2012 IEEE 12th international conference on data mining*, pages 924–929. IEEE, 2012. [3.2.3, 3.4]

[151] Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. Meritocratic fairness for infinite and contextual bandits. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 158–163, 2018. [3.2.3, 3.4]

[152] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *International Conference on Machine Learning*, pages 3150–3158. PMLR, 2018. [3.2.3, 3.4]

[153] Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 259–268, 2019. [3.2.3, 3.4]

[154] Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. The social

cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 230–239, 2019. [3.2.3, 3.4]

[155] Hadi Elzayn, Shahin Jabbari, Christopher Jung, Michael Kearns, Seth Neel, Aaron Roth, and Zachary Schutzman. Fair algorithms for learning in allocation problems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 170–179, 2019. [3.2.3, 3.4]

[156] Danielle Ensign, Sorelle A Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. Runaway feedback loops in predictive policing. In *Conference on fairness, accountability and transparency*, pages 160–171. PMLR, 2018. [3.2.3, 3.4]

[157] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012. [3.2.3, 3.4]

[158] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017. [3.2.3, 3.4]

[159] Faisal Kamiran and Toon Calders. Classifying without discriminating. In *2009 2nd international conference on computer, control and communication*, pages 1–6. IEEE, 2009. [3.4]

[160] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*, pages 35–50. Springer, 2012. [3.4]

[161] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013. [3.4]

[162] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016. [3.4, 3.3.3, B]

[163] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pages 962–970. PMLR, 2017. [3.4]

[164] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017. [3.4]

[165] Muhammad Bilal Zafar, Isabel Valera, Manuel Rodriguez, Krishna Gummadi, and Adrian Weller. From parity to preference-based notions of fairness in classification. *Advances in neural information processing systems*, 30, 2017. [3.4]

[166] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International conference on machine learning*, pages 60–69. PMLR, 2018. [3.4]

[167] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International conference on machine learning*, pages 2564–2572. PMLR, 2018. [3.4]

[168] Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. [3.4]

[169] L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 319–328, 2019. [3.4]

[170] Ilya M Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001. [3.2.4.1, 3.5]

[171] Andrea Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer physics communications*, 145(2):280–297, 2002. [3.2.4.1, 3.5]

[172] Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and Stefano Tarantola. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer physics communications*, 181(2):259–270, 2010. [3.2.4.1, 3.5]

[173] RI Cukier, CM Fortuin, Kurt E Shuler, AG Petschek, and J Ho Schaibly. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. i theory. *The Journal of chemical physics*, 59(8):3873–3878, 1973. [3.2.4.1, 3.5]

[174] Andrea Saltelli, Stefano Tarantola, and KP-S Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, 1999. [3.2.4.1, 3.5]

[175] Stefano Tarantola, Debora Gatelli, and Thierry Alex Mara. Random balance designs for the estimation of first order global sensitivity indices. *Reliability Engineering & System Safety*, 91(6):717–727, 2006. [3.2.4.1, 3.5]

[176] Elmar Plischke. An effective algorithm for computing global sensitivity indices (easi). *Reliability Engineering & System Safety*, 95(4):354–360, 2010. [3.2.4.1, 3.5]

[177] Jean-Yves Tissot and Clémentine Prieur. Bias correction for the estimation of sensitivity indices based on random balance designs. *Reliability Engineering & System Safety*, 107:205–213, 2012. [3.2.4.1, 3.5]

[178] Max D Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, 1991. [3.2.4.1, 3.5]

[179] A Saltelli. *Global Sensitivity Analysis: the Primer*. John Wiley & Sons, 2008. [3.2.4.1, 3.5]

[180] Francesca Campolongo, Jessica Cariboni, and Andrea Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental modelling & software*, 22(10):1509–1518, 2007. [3.2.4.1, 3.5]

[181] Emanuele Borgonovo. A new uncertainty importance measure. *Reliability Engineering & System Safety*, 92(6):771–784, 2007. [3.2.4.1, 3.5]

[182] Elmar Plischke, Emanuele Borgonovo, and Curtis L Smith. Global sensitivity measures from given data. *European Journal of Operational Research*, 226(3):536–550, 2013. [3.2.4.1, 3.5]

[183] Sergei Kucherenko and Shugfang Song. Derivative-based global sensitivity measures and their link with sobol'sensitivity indices. In *Monte Carlo and Quasi-Monte Carlo Methods: MCQMC, Leuven, Belgium, April 2014*, pages 455–469. Springer, 2016. [3.2.4.1, 3.5]

[184] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. [3.2.4.2, 3.6, 3.3.4, B]

[185] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [3.2.4.2, 3.6]

[186] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. [3.2.4.2, 3.6]

[187] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. [3.2.4.2, 3.6]

[188] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages

39–57. Ieee, 2017. [3.2.4.2, 3.6]

[189] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018. [3.2.4.2, 3.6]

[190] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019. [3.2.4.2, 3.6]

[191] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. [3.2.4.2, 3.6]

[192] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017. [3.2.4.2, 3.6]

[193] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, 2017. [3.2.4.2, 3.6]

[194] Suranjana Samanta and Sameep Mehta. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*, 2017. [3.2.4.2]

[195] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, 2018. [3.2.4.2, 3.6]

[196] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, 2018. [3.2.4.2, 3.6]

[197] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wen-

chang Shi. Deep text classification can be fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4208–4215, 2018. [3.2.4.2]

[198] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018. [3.2.4.2, 3.6]

[199] Xiaosen Wang, Hao Jin, and Kun He. Natural language adversarial attack and defense in word level. 2019. [3.2.4.2, 3.6]

[200] Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3601–3608, 2020. [3.2.4.2, 3.6]

[201] Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, 2020. [3.2.4.2, 3.6]

[202] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, 2020. [3.2.4.2, 3.6]

[203] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, 2020. [3.2.4.2, 3.6]

[204] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*,

pages 2847–2856, 2018. [3.2.4.2, 3.6]

[205] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In *International conference on machine learning*, pages 1115–1124. PMLR, 2018. [3.2.4.2, 3.6]

[206] Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *International Conference on Learning Representations (ICLR)*, 2019. [3.2.4.2, 3.6]

[207] Moustapha M Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. *Advances in neural information processing systems*, 30, 2017. [3.2.4.2]

[208] Jonas Rauber and Matthias Bethge. Fast differentiable clipping-aware normalization and rescaling. *arXiv preprint arXiv:2007.07677*, 2020. [3.2.4.2, 3.6]

[209] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *International Conference on Machine Learning*, pages 3866–3876. PMLR, 2019. [3.2.4.2, 3.6]

[210] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018. [3.6]

[211] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016. [3.6]

[212] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2022. [3.6]

[213] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recog-

nition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016. [3.6]

[214] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. In *International Conference on Learning Representations*, 2022. [3.6]

[215] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *CVPR Workshops*, volume 2, 2017. [3.6]

[216] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning, 2017. In *ACM Asia Conference on Computer and Communications Security*, 2016. [3.6]

[217] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017. [3.6]

[218] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017. [3.6]

[219] Nicholas Carlini, Guy Katz, Clark Barrett, and David L Dill. Provably minimally-distorted adversarial examples. *arXiv preprint arXiv:1709.10207*, 2017. [3.6]

[220] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. [3.6]

[221] L Schott, J Rauber, M Bethge, and W Brendel. Towards the first adversarially robust neural network model on mnist. In *Seventh International Conference on Learning Representations (ICLR 2019)*, pages 1–16, 2019. [3.6]

[222] Shi Feng, Eric Wallace, Alvin Grissom II, Pedro Rodriguez, Mohit Iyyer,

and Jordan Boyd-Graber. Pathologies of neural models make interpretation difficult. In *Empirical Methods in Natural Language Processing*, 2018. [3.6]

[223] Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 43–49. IEEE, 2018. [3.6]

[224] J Li, S Ji, T Du, B Li, and T Wang. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*, 2019. [3.6]

[225] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE, 2018. [3.6]

[226] Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. Adversarial examples for natural language classification problems. 2018. [3.6]

[227] Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. *Advances in neural information processing systems*, 32, 2019. [3.6]

[228] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pages 1277–1294. IEEE, 2020. [3.6]

[229] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1085–1097, 2019. [3.6]

[230] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020. [3.6]

[231] Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. It's mor-

phin'time! combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2920–2935, 2020. [3.6]

[232] Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158, 2012. [3.3.2, B]

[233] Véronne Yepmo, Grégory Smits, and Olivier Pivert. Anomaly explanation: A review. *Data & Knowledge Engineering*, 137:101946, 2022. [4.1]

[234] Egawati Panjei, Le Gruenwald, Eleazar Leal, Christopher Nguyen, and Shejuti Silvia. A survey on outlier explanations. *The VLDB Journal*, 31(5):977–1008, 2022. [4.1]

[235] Avraam Bardos, Panagiotis Doupidis, Thanasis Kotsiopoulos, Thanasis Vafeiadis, Alexandros Nizamis, Heino Geerdens, Dimosthenis Ioannidis, Konstantinos Votis, Dimitrios Tzovaras, and Panagiotis Sarigiannidis. Anomaly detection in industrial processes: Supervised vs. unsupervised learning and the role of explainability. *Unsupervised Learning and the Role of Explainability*, 2024. [4.1]

[236] Sreeraj Rajendran, Wannes Meert, Vincent Lenders, and Sofie Pollin. Saife: Unsupervised wireless spectrum anomaly detection with interpretable features. In *2018 IEEE international symposium on dynamic spectrum access networks (DySPAN)*, pages 1–9. IEEE, 2018. [4.1.1]

[237] Magnus Gribbestad, Muhammad Umair Hassan, Ibrahim A Hameed, and Kelvin Sundli. Health monitoring of air compressors using reconstruction-based deep learning for anomaly detection with increased transparency. *Entropy*, 23(1):83, 2021. [4.1.1]

[238] Tung Kieu, Bin Yang, Chenjuan Guo, Christian S Jensen, Yan Zhao, Feiteng Huang, and Kai Zheng. Robust and explainable autoencoders for unsupervised time series outlier detection. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 3038–3050. IEEE, 2022. [4.1.1]

[239] Diana Laura Aguilar, Miguel Angel Medina-Pérez, Octavio Loyola-Gonzalez, Kim-Kwang Raymond Choo, and Edoardo Bucheli-Susarrey. Towards an interpretable autoencoder: A decision-tree-based autoencoder and its application in anomaly detection. *IEEE transactions on dependable and secure computing*, 20(2):1048–1059, 2022. [4.1.1]

[240] Michael Alexander Zenkl-Galaz, Octavio Loyola-González, and Miguel Angel Medina-Pérez. Iogod: An interpretable outlier generation-based outlier detector for categorical databases. *Expert Systems with Applications*, 195:116570, 2022. [4.1.1]

[241] Yuehua Huang, Wenfen Liu, Song Li, Ying Guo, and Wen Chen. Interpretable single-dimension outlier detection (isod): An unsupervised outlier detection method based on quantiles and skewness coefficients. *Applied Sciences*, 14(1):136, 2023. [4.1.1]

[242] Ashima Chawla, Paul Jacob, Saman Feghhi, Devashish Rughwani, Sven van der Meer, and Sheila Fallon. Interpretable unsupervised anomaly detection for ran cell trace analysis. In *2020 16th International Conference on Network and Service Management (CNSM)*, pages 1–5. IEEE, 2020. [4.1.1]

[243] Mattia Carletti, Matteo Terzi, and Gian Antonio Susto. Interpretable anomaly detection with diffi: Depth-based feature importance of isolation forest. *Engineering Applications of Artificial Intelligence*, 119:105730, 2023. [4.1.1]

[244] Botao An, Shibin Wang, Fuhua Qin, Zhibin Zhao, Ruqiang Yan, and Xuefeng Chen. Adversarial algorithm unrolling network for interpretable mechanical anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. [4.1.1]

[245] Yifei Yang, Peng Wang, Xiaofan He, and Dongmian Zou. Gram: An interpretable approach for graph anomaly detection using gradient attention maps. *Neural Networks*, page 106463, 2024. [4.1.1]

[246] Huimin Zhao, Yandong Wu, and Wu Deng. An interpretable dynamic inference system based on fuzzy broad learning. *IEEE Transactions on Instrumentation and Measurement*, 2023. [4.1.1]

[247] Man Zhou, Lansheng Han, and Jingpei Wang. Physical invariant subspace based unsupervised anomaly detection for internet of vehicles. *IEEE Transactions on Intelligent Vehicles*, 2024. [4.1.1]

[248] ST Roche, Q Bayer, BT Carlson, WC Ouligian, P Serhiayenka, J Stelzer, and TM Hong. Nanosecond anomaly detection with decision trees and real-time application to exotic higgs decays. *Nature Communications*, 15(1):3527, 2024. [4.1.1]

[249] Albert Rachwał, Paweł Karczmarek, Alicja Rachwał, and Rafał Stęgierski. Isolation forest with exclusion of attributes based on shapley index. *IEEE Access*, 2024. [4.1.1]

[250] Liuqing Yang, Sergey Fomel, Shoudng Wang, Xiaohong Chen, Yaoguang Sun, and Yangkang Chen. Interpretable unsupervised learning framework for multi-dimensional erratic and random noise attenuation. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. [4.1.1]

[251] Bixuan Gao, Xiangyu Kong, Shangze Li, Yi Chen, Xiyuan Zhang, Ziyu Liu, and Weijia Lv. Enhancing anomaly detection accuracy and interpretability in low-quality and class imbalanced data: A comprehensive approach. *Applied Energy*, 353:122157, 2024. [4.1.1]

[252] Maximilian Schmidt and Marko Simic. Normalizing flows for novelty detection in industrial time series data. *arXiv preprint arXiv:1906.06904*, 2019. [4.1.2, 4.1]

[253] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2828–2837, 2019. [4.1.2, 4.1]

[254] Yihong Ma, Md Nafee Al Islam, Jane Cleland-Huang, and Nitesh V Chawla. Detecting anomalies in small unmanned aerial systems via graphical normalizing flows. *IEEE Intelligent Systems*, 2023. [4.1.2, 4.1]

[255] Xigang Zhao, Peng Liu, Saïd Mahmoudi, Sahil Garg, Georges Kaddoum,

and Mohammad Mehedi Hassan. Ddanf: Deep denoising autoencoder normalizing flow for unsupervised multivariate time series anomaly detection. *Alexandria Engineering Journal*, 108:436–444, 2024. [4.1.2, 4.1]

[256] Madson LD Dias, César Lincoln C Mattos, Ticiana LC da Silva, José Antônio F de Macedo, and Wellington CP Silva. Anomaly detection in trajectory data with normalizing flows. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020. [4.1.2, 4.1]

[257] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021. [4.1.2, 4.1]

[258] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 98–107, 2022. [4.1.2, 4.1]

[259] Jordan J Bannister, Matthias Wilms, J David Aponte, David C Katz, Ophir D Klein, Francois PJ Bernier, Richard A Spritz, Benedikt Hallgrímsson, and Nils D Forkert. Detecting 3d syndromic faces as outliers using unsupervised normalizing flow models. *Artificial Intelligence in Medicine*, 134:102425, 2022. [4.1.2, 4.1]

[260] Yuzhong Zhao, Qiaoqiao Ding, and Xiaoqun Zhang. Ae-flow: Autoencoders with normalizing flows for medical images anomaly detection. In *The Eleventh International Conference on Learning Representations*, 2023. [4.1.2, 4.1]

[261] Or Hirschorn and Shai Avidan. Normalizing flows for human pose anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13545–13554, 2023. [4.1.2, 4.1]

[262] Ruituo Wu, Yang Chen, Jian Xiao, Bing Li, Jicong Fan, Frédéric Dufaux, Ce Zhu, and Yipeng Liu. Da-flow: Dual attention normalizing flow for skeleton-based video anomaly detection. *arXiv preprint arXiv:2406.02976*, 2024. [4.1.2, 4.1]

[263] Wei Liu, Jun Chen, et al. Adaptive graph convolution embedding normalizing flow for video anomaly detection. *Preprint*, 06 2024. [4.1.2, 4.1]

[264] Naji Najari, Samuel Berlemont, Grégoire Lefebvre, Stefan Duffner, and Christophe Garcia. Robust variational autoencoders and normalizing flows for unsupervised network anomaly detection. In *International Conference on Advanced Information Networking and Applications*, pages 281–292. Springer, 2022. [4.1.2, 4.1]

[265] Rui Cao, Shijie Xue, Jindong Li, Qi Wang, and Yi Chang. Fanfold: Graph normalizing flows-driven asymmetric network for unsupervised graph-level anomaly detection. *arXiv preprint arXiv:2407.00383*, 2024. [4.1.2, 4.1]

[266] Nishant Kumar, Pia Hanfeld, Michael Hecht, Michael Bussmann, Stefan Gumhold, and Nico Hoffmann. Inflow: Robust outlier detection utilizing normalizing flows. *arXiv preprint arXiv:2106.12894*, 2021. [4.1.2, 4.1]

[267] Bodo Rosenhahn and Christoph Hirche. Quantum normalizing flows for anomaly detection. *arXiv preprint arXiv:2402.02866*, 2024. [4.1.2, 4.1]

[268] Ioana Ciuca and Yuan-Sen Ting. Unsupervised learning for stellar spectra with deep normalizing flows. *arXiv preprint arXiv:2207.02785*, 2022. [4.1.2, 4.1]

[269] Samuel von Baußnern, Johannes Otterbach, Adrian Loy, Mathieu Salzmann, and Thomas Wollmann. Daain: Detection of anomalous and adversarial input using normalizing flows. *arXiv preprint arXiv:2105.14638*, 2021. [4.1]

[270] Xiaorun Li, Shaoqi Yu, Shuhan Chen, and Liaoying Zhao. Normalizing flow-based probability distribution representation detector for hyperspectral anomaly detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:4885–4896, 2022. [4.1]

[271] MyeongAh Cho, Taeoh Kim, Woo Jin Kim, Suhwan Cho, and Sangyoun Lee. Unsupervised video anomaly detection via normalizing flows with implicit latent features. *Pattern Recognition*, 129:108703, 2022. [4.1]

[272] Li-Ling Chiu and Shang-Hong Lai. Self-supervised normalizing flows for image anomaly detection and localization. In *Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition, pages 2926–2935, 2023. [4.1]

[273] Nishant Kumar, Siniša Šegvić, Abouzar Eslami, and Stefan Gumhold. Normalizing flow based feature synthesis for outlier-aware object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5156–5165, 2023. [4.1]

[274] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. [4.2.2, B]

[275] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017. [4.2.2.2]

[276] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. [4.2.2.4]

[277] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2023. [4.3.2.4, 4.3.2.4]

[278] Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. *Advances in neural information processing systems*, 11, 1998. [4.3.2.4]

[279] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009. [4.3.2.4]

[280] Tharindu R Bandaragoda, Kai Ming Ting, David Albrecht, Fei Tony Liu, and Jonathan R Wells. Efficient anomaly detection by isolation using nearest neighbour ensemble. In *2014 IEEE International conference on data mining workshop*, pages 698–705. IEEE, 2014. [4.3.2.4]

[281] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *2020 IEEE international conference on data mining (ICDM)*, pages 1118–1123. IEEE, 2020. [4.3.2.4]

[282] Mahito Sugiyama and Karsten Borgwardt. Rapid distance-based outlier detection via sampling. *Advances in neural information processing systems*, 26, 2013. [4.3.2.4]

[283] Adam Goodge, Bryan Hooi, See-Kiong Ng, and Wee Siong Ng. Lunar: Unifying local outlier detection methods via graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6737–6745, 2022. [4.3.2.4]

[284] Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 61–75. Springer, 2007. [4.3.2.4]

[285] Andreas Arning, Rakesh Agrawal, and Prabhakar Raghavan. A linear method for deviation detection in large databases. In *KDD*, volume 1141, pages 972–981, 1996. [4.3.2.4]

[286] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017. [4.3.2.4]

[287] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018. [4.3.2.4]

[288] M Scott, Lee Su-In, et al. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30:4765–4774, 2017. [4.4.7.7]